ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

INGENIERO ELECTROMECÁNICO

APLICACIÓN WEB BASADA EN HTML5 PARA INTERFAZ CON UN SISTEMA DOMÓTICO X10

Autor: Alberto Arce Arroyo
Director: José Daniel Muñoz Frías

Madrid
Julio 2014

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN ACCESO *ABIERTO* (*RESTRINGIDO*) DE DOCUMENTACIÓN

1º. Declaración de la autoría y acreditación de la misma.

El autor D. Alberto Arce Arroyo , como alumno de la UNIVERSIDAD PONTIFICIA COMILLAS (COMILLAS), **DECLARA**

que es el titular de los derechos de propiedad intelectual, objeto de la presente cesión, en relación con la obra proyecto fin de carrera[1], que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual como titular único o cotitular de la obra.

En caso de ser cotitular, el autor (firmante) declara asimismo que cuenta con el consentimiento de los restantes titulares para hacer la presente cesión. En caso de previa cesión a terceros de derechos de explotación de la obra, el autor declara que tiene la oportuna autorización de dichos titulares de derechos a los fines de esta cesión o bien que retiene la facultad de ceder estos derechos en la forma prevista en la presente cesión y así lo acredita.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad y hacer posible su utilización de *forma libre y gratuita* (*con las limitaciones que más adelante se detallan*) por todos los usuarios del repositorio y del portal e-ciencia, el autor **CEDE** a la Universidad Pontificia Comillas de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución, de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra (a) del apartado siguiente.

3º. Condiciones de la cesión.

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia, el repositorio institucional podrá:

(a) Transformarla para adaptarla a cualquier tecnología susceptible de incorporarla a internet; realizar adaptaciones para hacer posible la utilización de la obra en formatos electrónicos, así

[1] Especificar si es una tesis doctoral, proyecto fin de carrera, proyecto fin de Máster o cualquier otro trabajo que deba ser objeto de evaluación académica

como incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.

(b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato. .

(c) Comunicarla y ponerla a disposición del público a través de un archivo abierto institucional, accesible de modo libre y gratuito a través de internet.[2]

(d) Distribuir copias electrónicas de la obra a los usuarios en un soporte digital. [3]

4º. Derechos del autor.

El autor, en tanto que titular de una obra que cede con carácter no exclusivo a la Universidad por medio de su registro en el Repositorio Institucional tiene derecho a:

a) A que la Universidad identifique claramente su nombre como el autor o propietario de los derechos del documento.

b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.

c) Solicitar la retirada de la obra del repositorio por causa justificada. A tal fin deberá ponerse en contacto con el vicerrector/a de investigación (curiarte@rec.upcomillas.es).

d) Autorizar expresamente a COMILLAS para, en su caso, realizar los trámites necesarios para la obtención del ISBN.

d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

---

[2] En el supuesto de que el autor opte por el acceso restringido, este apartado quedaría redactado en los siguientes términos:

(c) Comunicarla y ponerla a disposición del público a través de un archivo institucional, accesible de modo restringido, en los términos previstos en el Reglamento del Repositorio Institucional

[3] En el supuesto de que el autor opte por el acceso restringido, este apartado quedaría eliminado.

5º. Deberes del autor.

El autor se compromete a:

a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.

b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.

c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.

d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

a) Deberes del repositorio Institucional:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.

- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.

- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.

b) Derechos que se reserva el Repositorio institucional respecto de las obras en él registradas:

- retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 11 de 7 de 2014.

ACEPTA

Fdo.....

Autorizada la entrega del proyecto del alumno/a:

Alberto Arce Arroyo

.....

EL DIRECTOR DEL PROYECTO

José Daniel Muñoz Frías

Fdo.: ..... Fecha: 11/ 7/ 2014

V° B° del Coordinador de Proyectos

Álvaro Sánchez Miralles

Fdo.: ..... Fecha: ...../ ...../ .....

SCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

INGENIERO ELECTROMECÁNICO

APLICACIÓN WEB BASADA EN HTML5 PARA INTERFAZ CON UN SISTEMA DOMÓTICO X10

Autor: Alberto Arce Arroyo
Director: José Daniel Muñoz Frías

Madrid
Julio 2014

APLICACIÓN WEB BASADA EN HTML5 PARA INTERFAZ CON UN SISTEMA DOMÓTICO X10

Autor: Arce Arroyo, Alberto.

Director: Muñoz Frías, José Daniel.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas.

RESUMEN DEL PROYECTO

1.- INTRODUCCIÓN

El proyecto consiste en desarrollar una aplicación domótica de bajo coste donde se monitoricen y controlen algunos elementos que se presentan en un hogar, estos elementos pueden ser radiadores, persianas, luces, etc.

La aplicación web será accesible desde el exterior y representará, por medio de una interfaz sencilla e intuitiva, la planta de la vivienda con los diferentes elementos y un registro histórico de temperaturas.

Dicha aplicación correrá sobre un servidor web de bajo coste, en este caso, un dispositivo de propósito general, la Raspberry Pi, que se comunicará con un Arduino mediante el protocolo de conexión SPI, dicha comunicación establecerá comunicación entre la aplicación y la vivienda.

2.- PLANTEAMIENTO DEL PROBLEMA

Se plantea crear una aplicación web, con las ventajas de tener acceso externo (aplicación web adaptada a móviles), poder monitorizar y controlar nuestra vivienda desde un entorno seguro (Cifrado de conexión), además de contar con las ventajas comunes a instalaciones domóticas ya existentes: confort y ahorro energético.

3.- ESTADO DE LA TÉCNICA

Centraremos la información de este apartado exclusivamente en las telecomunicaciones ofrecidas por los fabricantes de sistemas domóticos, en concreto valoramos accesibilidad, seguridad y las diferentes formas de control de los elementos eléctricos, ver tablas 1 y 2.

	INTERNET		
	PLANTA	ORDENES	SEGURIDAD
DOMOTICA VIVA	X	X	CIFRADO
EUROCABLE		X	USUARIO/CONTRASEÑA
DOMOTICUS		X	USUARIO/CONTRASEÑA
INDOMO		X	USUARIO/CONTRASEÑA

Tabla 1.Tecnologías existentes, internet.

	TELEFONO			CONTROL
	AVISO	SMS	APP	VOZ
DOMOTICA VIVA				
EUROCABLE	X	X		
DOMOTICUS		X	X	X
INDOMO			X	X

Tabla 2.- Tecnologías existentes, teléfono y voz.

En el proyecto queremos reunir las siguientes características: seguridad y accesibilidad externa desde cualquier dispositivo, todo para crear un producto con las mismas competencias con respecto a las empresas actuales, por lo que ciframos las comunicaciones y creamos una aplicación web adaptada a móviles, sin necesidad de envió de sms o control por voz.

4.- OBJETO DEL PROYECTO

El primer objetivo se presenta como la representación de la planta de la casa, por lo que la aplicación web está basada en HTLM5, con el fin de utilizar su nueva aplicación denominada CANVAS (lienzo, traducido del inglés) que permite la renderización scriptable dinámica de formas 2D.

El segundo objetivo es una representación por medio de gráficas, de un registro histórico de temperaturas, con el fin de mejorar la experiencia de monitorización.

5.- METODOLOGÍA

El método para abordar el proyecto se divide en cuatro partes principales:

1.- Configuración Raspberry Pi

2.- Instalación del servidor

3.- Configuración del router

4.- Creación de base de datos

5.- Desarrollo de la programación

5.1.- Configuración Raspberry Pi

Acceso interno

Instalado el sistema operativo Raspbian en la Raspberry, la primera configuración debe ser SSH, para poder acceder a la Raspberry Pi de forma remota sin necesidad de tener conectado un TV o monitor.

Para tal fin es necesario establecer una ip estática, modificando la configuración de red**.**

Acceso externo

Para acceder desde el exterior debemos conocer nuestra ip, dado que tenemos una configuración ip dinámica, cada vez que reiniciamos ésta cambia.

Vamos a utilizar el servicio de DNS dinámica de No-IP, que permite identificar tu PC con un nombre de dominio fácil de recordar, como en nuestro caso, domo.no-ip.biz en lugar de recordar la ip sin importar si está cambia y así poder acceder desde el exterior.

5.2.- Instalación y configuración del servidor

Una vez configurada la Raspberry Pi empezaremos a instalar el servidor Web. Nuestro servidor web dispondrá de los elementos básicos:

Un servidor web o software que permita a nuestro ordenador alojar páginas web dinámicas. En nuestro caso utilizaremos Apache, un software de código abierto y libre que funciona en cualquier sistema operativo.

El entorno PHP que nos permiten establecer comunicación entre la página y el servidor.

Un sistema de gestión de bases de datos con las que los programas en PHP interactuarán. En nuestro caso utilizaremos MySQL un sistema de bases de datos relacional que se distribuye bajo licencia GNU GPL para proyectos que se distribuyan bajo esa misma licencia.

Un administrador de MySQL, PhpMyAdmin, a través de páginas web, utilizando Internet, que permite crear y manejar bases de datos con una interfaz más sencilla, PhpMyAdmin.

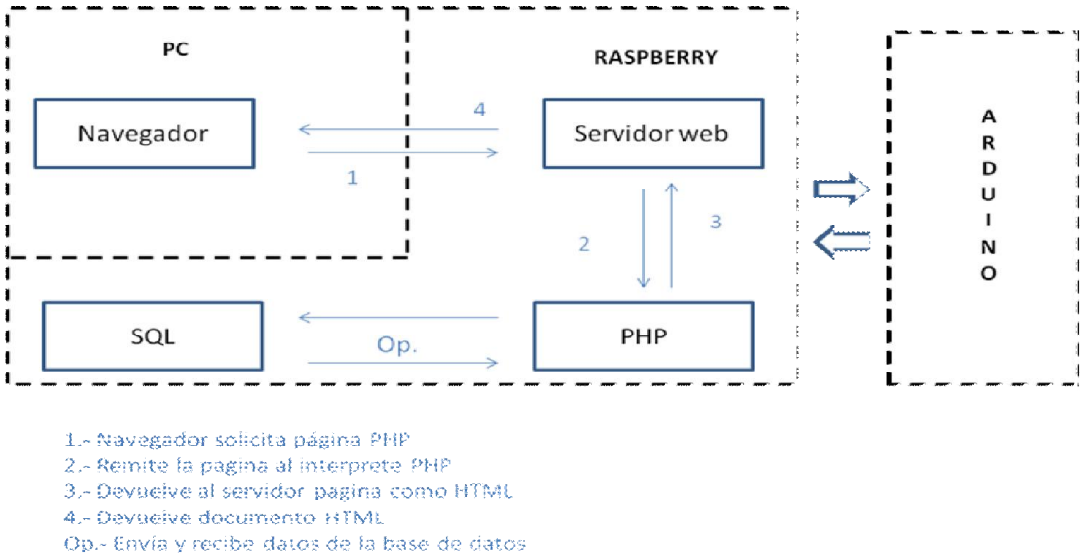

Figura 1. Funcionamiento del servidor.

Puesto que una de las principales cuestiones que se planteaban, era el problema de la seguridad en internet, se opto por la decisión de cifrar las comunicaciones como medida de mayor garantía ante dicha inseguridad, incorporando certificados de cifrado con duración de 3 años,

5.3.- Configuración del router

Para terminar de configurar el acceso externo hay que abrir ciertos puertos del router, ya que un servidor es una aplicación que está diseñada para enviar datos hacia aquellos usuarios que se los solicitan.

PUERTOS	SERVICIO
22	SSH
21	FTP
80	HTTP
5900-5910	VNC

Tabla 3. Puertos abiertos router.

En mi caso directamente aplique la configuración DMZ, esta herramienta consiste en determinar una zona que actúa como una salida a internet con todos los puertos abiertos.

5.4.- Creación de tablas en base de datos

Una base de datos es la herramienta usada para guardar la información actual de la casa, así como también puede ser usada para guardar el registro histórico de temperaturas que se produce en cada habitación. Su uso se hace muy sencillo con la herramienta PhpMyAdmin.

5.5.- Programación

En un primer instante se define la arquitectura de la pagina web, es decir la forma de presentar el contenido, para ello he usado una plantilla gratuita, denominada "Response", adaptada a móviles.

El esquema de la planta de la vivienda fue el siguiente paso, la herramienta utilizada en este caso es CANVAS, un elemento de HTML5 que permite generar gráficos dinámicamente.

Habiendo hecho un primer contacto con respecto al primer objetivo, el segundo objetivo, el histograma, era el siguiente paso a realizar. Para la representación de las temperaturas se valora como la mejor opción, dada la facilidad de implementación, así como una mejor experiencia interactiva con el usuario, la aplicación de Google denominada Google Charts.

Sabiendo cómo crear un gráfico de temperaturas para cada habitación, es momento de representar todas las gráficas en una misma página, se opto por una representación

oculta nada más cargar la página que mostrase u ocultase cada una de las gráficas al hacer click en la opción correspondiente a cada habitación, de esta manera conseguimos una visualización más clara y accesible al usuario, usando iframes.

En este momento la página está desarrollada en HTML, es estática, por lo que hay que desarrollar la programación PHP para usar contenido dinámico, para ello hay que ser capaz de interactuar con la base de datos que simula el comportamiento de la vivienda.

a. Realizar consultas para representar la situación de la vivienda.

b. Mostrar datos de dichas consultas en los histogramas de temperatura.

c. Actualizar los datos al querer cambiar el estado de los radiadores.

En la actualización de los radiadores, el funcionamiento que se quiere representar es similar al de un interruptor, pero para simular su comportamiento hay que llamar desde HTML código PHP, pero no es posible.

La forma más "natural" de comunicación cliente-servidor es utilizar formularios de envío con el método POST.

6.- RESULTADOS

Se ha conseguido una aplicación eficaz y eficiente, a la par que fiable y estable, con una curva de aprendizaje mínima y con la funcionalidad más utilizada a la vista, evitando complejos procedimientos para su uso.

WEB APPLICATION BASED ON HTML5 INTERFACING TO HOME AUTOMATION SYSTEM X10

Author: Arce Arroyo, Alberto.

Director: Muñoz Frías, José Daniel.

Collaborating Organization: ICAI – Universidad Pontificia Comillas.

PROJECT SUMMARY

1.- INTRODUCTION

The project is to develop a low cost home automation application where you can monitor and control electric household items, these items can be radiators, blinds, lights.

The web application will be accessible from the outside and represent, through a simple and intuitive interface, the floor of the house with the different household items, and a historical temperatures record, represented by a line graph.

This application will run on a web server housing web-low cost, in this case, a general purpose device, the Raspberry Pi, that will communicate with an Arduino connected via SPI.

2.- PROBLEM

To create a web application, with the advantages of having external access (web application adapted to mobile phones), to monitor and control our home from a secure (encryption on the web server), in addition to the common advantages for automation installations existing, comfort and energy savings.

3.- PRIOR ART

Focus the information in this section exclusively offered by telecommunications manufacturers automation systems, see Table 1 and 2.

	INTERNET		
	PLANTA	ORDENES	SEGURIDAD
DOMOTICA VIVA	X	X	CIFRADO
EUROCABLE		X	USUARIO/CONTRASEÑA
DOMOTICUS		X	USUARIO/CONTRASEÑA
INDOMO		X	USUARIO/CONTRASEÑA

Table 4.Tecnologies, internet.

	TELEFONO			CONTROL
	AVISO	SMS	APP	VOZ
DOMOTICA VIVA				
EUROCABLE	X	X		
DOMOTICUS		X	X	X
INDOMO			X	X

Table 5.- Technologies, mobile phone.

We want to have the following characteristics: security and external access from any device, all to create a product with the same powers with respect to existing businesses, so we encrypt the server and create a web application adapted to mobile without sms or voice control.

4.- OBJECTIVES

Because our first objective, representing the floor of the house, we need a web application based on HTML5, to use CANVAS, allowing rendering scriptable dynamics of 2D shapes.

The second objective is a representation of a historical temperature record, information log temperatures represented by a line graph of type.

5.- PROCESS

The method to approach the project is divided into four main parts:

1.- Raspberry Pi Settings

2.- Installing the server & Server configuration

3.- Router configuration

4.- Database

5.- Develop programming

5.1.- Raspberry Pi Settings

Internal access

Raspbian operating system is installed on the Raspberry, the first configuration should be SSH to access the Raspberry Pi remotely without having connected a TV or monitor.

To this end it is necessary to set a static ip, changing network settings**.**

External access

To access from the outside must know our ip, since we have a dynamic IP configuration, reboot each time is changed.

We will use the dynamic DNS service No-IP, which identifies your PC with a domain name easy to remember, as in our case, domo.no-ip.biz instead of remembering the IP no matter if you are changing.

5.2.- Installing the server & Server configuration

Install the Web server, our web server will have the follow basic elements:

A web server or software that allows your computer to accommodate dynamic web pages. In our case we use Apache, an open source software and open source that works on any operating system.

The PHP environment consisting of a set of programs that allow us to apply for programs written in this programming language from a web server, which is then displayed in the browser in HTML.

A management system database with PHP programs interact. In our case we use a set of MySQL relational database that is distributed under GNU GPL for projects to be distributed under the same license.

An administrator of MySQL, PhpMyAdmin, through web pages, using the Internet, which allows you to create and manage databases with a simpler interface, PhpMyAdmin.

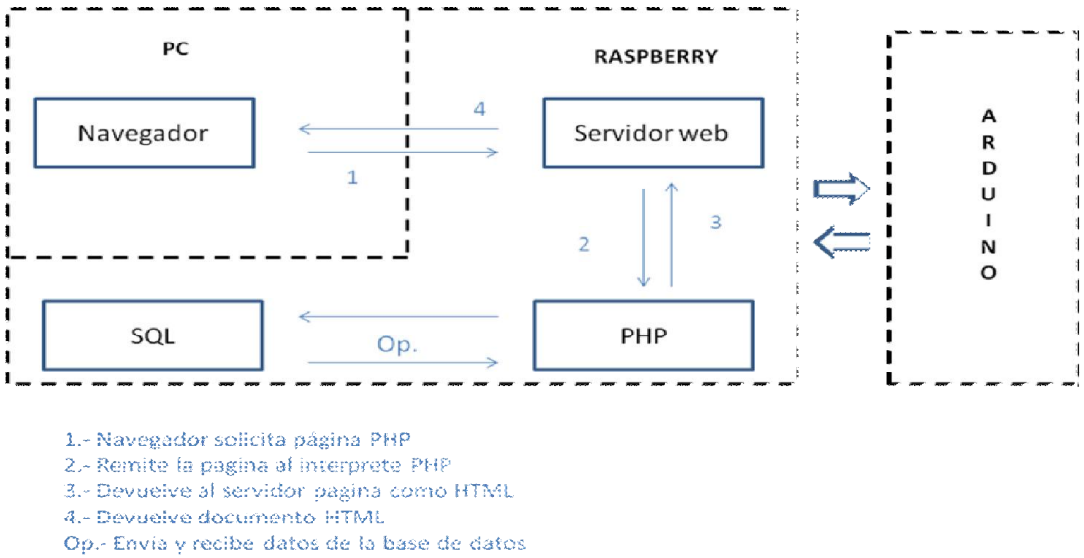

Figure 2. Server Operation.

The problem of Internet security, we opted for the decision to encrypt the server as a measure of increased security with such uncertainty, incorporating encryption certificates for a period of 3 years.

5.3.- Router configuration

To finish external access configuration, we must open certain ports on the router, and a server is an application that is designed to send data to those users who request them.

PUERTOS	SERVICIO
22	SSH
21	FTP
80	HTTP
5900-5910	VNC

Table 6. Open ports, router.

I apply the DMZ configuration, this tool is to identify an area that serves as an outlet for internet with all open ports.

5.4.- Database

A database is usefull tool to save the current house information, and can also be used to store the historical record of temperatures occurring in each room. Its use is very simple with the PhpMyAdmin tool.

5.5.- Programming

The architecture of the website is defined, this a free template for "Response" was downloaded adapted to mobile.

With the template adapted to us, develop the scheme of the plant housing was the next step, the tool used in this case is CANVAS, an element of HTML5 that allows to generate graphics dynamically.

With a first contact with the first objective, the second objective, the histogram, the next step was to perform. For the representation of the temperatures is valued as the best option given the ease of implementation as well as a better interactive experience to the user, the application of Google called Google Charts.

Knowing how to create a graph of temperatures for each room, it's time to represent all graphs on the same page, we opted for a hidden representation after loading the page that would show or concealed each of the graphs by clicking on the appropriate option each room, so we get a clearer and more accessible to the user display, we use iframes.

Right now the site is developed in HTML, is static, so we need to develop PHP programming to use dynamic content, for this we must be able to interact with the database that simulates the behavior of housing.

a. Database querying.

b. Displaying Data on temperature histograms.

c. Update, change radiator status.

Updating of the radiators, the operation being represented is similar to a switch.

To upgrade the database was necessary to call PHP from HTML code, but it is not possible.

The most "natural" form of client-server communication is to use submission forms with method POST.

6.- RESULTS

We has got an effective and efficient web server application, at the same time reliable and stable, with a minimal learning curve and the most used functionality in sight, avoiding complex procedures for use.

Índice de la memoria

Parte I Memoria ***2***

Capítulo 1 Introducción ***3***

1.1 Estudio de los trabajos existentes / tecnologías existentes **3**

1.1.1 DOMOTICA VIVA 4

1.1.2 EUROCABLE, S.L. 4

1.1.3 DOMOTICUS 4

1.1.4 INDOMO 5

1.1.5 Resumen 5

1.1.6 Valoración 5

1.2 Estudio de los diferentes dispositivos de propósito general **6**

1.2.1 Definición de los requisitos 6

1.2.2 Alternativas 6

1.2.3 Valoración 10

1.3 Motivación del proyecto **10**

1.4 Objetivos **11**

1.5 Metodología / Solución desarrollada **12**

1.5.1 Configuración la Raspberry pi I 12

1.5.2 Instalación del servidor 13

1.5.3 Configuración la Raspberry Pi II 15

1.5.4 Configuración del ROUTER 18

1.5.5 Desarrollar la programación I 19

1.5.6 Creación de la base de datos 25

1.5.7 Desarrollo de la programación II 27

1.6 Recursos / herramientas empleadas **30**

1.6.1 Raspberry Pi 30

1.6.2 Raspbian 31

1.6.3 Apache 31
1.6.4 PHP 31
1.6.5 Codecademy 31
1.6.6 canvas 32
1.6.7 Google Charts API 32
1.6.8 MySQL 32
1.6.9 PhpMyAdmin 33

Capítulo 2 Conclusiones 34

Capítulo 3 Futuros desarrollos 35

3.1 Conexión SPI 35

3.1.1 Introducción 35
3.1.2 Instalación y uso de librería WiringPi 36
3.1.3 Configuración del acceso a MySQL desde C 37
3.1.4 Comunicación con MySQL 37

Capítulo 4 Bibliografía 40

Parte II Estudio económico 42

Capítulo 1 Estudio económico 43

Parte III Manual de usuario 44

1.1 Manual de usuario aplicación web 45

1.1.1 Diseño 45
1.1.2 Descripción de contenidos 45

Parte IV Código fuente 47

Capítulo 1 Código fuente 48

Parte V Ánexos ..... ¡Error! Marcador no definido.

Capítulo 1 Conocimientos de Programación 65

Índice de figuras

Figura 1. Carambola 6
Figura 2. Cubieboard 7
Figura 3. HackBerryA10 8
Figura 4. Raspberry Pi 9
Figura 5. Funcionamiento servidor. 14
Figura 6. Ejemplo del cifrado 17
Figura 7. Configuración DMZ del router. 19
Figura 8. Representación de la planta de la vivienda. 21
Figura 9. Histograma de temperatura. 24
Figura 10. PhpMyAdmin 26
Figura 11. Variables tabla Radiadores. 26
Figura 12. Variables tabla habitacion1 27
Figura 13. Representación de temperaturas con datos BBD 28
Figura 14. Representación de habitación con datos BBD 28
Figura 15. Representación de interruptores 29
Figura 17. Conexión SPI 36
Figura 18. Planta 46
Figura 19. Histograma 46
Figura 20. Estructura archivos de la carpeta /www 48

Índice de tablas

Tabla 1. Tecnologías existentes, internet. 5

Tabla 2. Tecnologías existentes, teléfono y voz. 5

Tabla 3. Puertos abiertos router. 19

Parte I MEMORIA

Capítulo 1 INTRODUCCIÓN

En este proyecto se desarrollará una aplicación web basada en HTML 5 para hacer de interfaz con un sistema domótico X10. La aplicación web se desarrollará en un sistema empotrado de bajo coste como Raspberry Pi, la cual se comunicará con un controlador X10 basado en Arduino.

El proyecto consiste en desarrollar una aplicación domótica de bajo coste donde se monitoricen y controlen algunos elementos que se presentan en un hogar, estos elementos pueden ser radiadores, persianas, luces.

La aplicación web, será accesible desde el exterior y representara, por medio de una interfaz sencilla e intuitiva, la planta de la vivienda con los diferentes elementos, y un registro histórico de temperaturas representado por un gráfico de líneas.

Dicha aplicación correrá sobre un servidor web de bajo coste, en este caso, un dispositivo de propósito general, la Raspberry Pi, que se comunicará con un Arduino mediante el protocolo de conexión SPI para establecer comunicación aplicación web-vivienda.

1.1 ESTUDIO DE LOS TRABAJOS EXISTENTES / TECNOLOGÍAS EXISTENTES

En este apartado nos centraremos exclusivamente en las telecomunicaciones ofrecidas por los fabricantes de sistemas domóticos en el mercado actual.

1.1.1 DOMOTICA VIVA

- Desde cualquier PC conectado a la red LAN CAT 5 o por WiFi, mediante plano de la vivienda, se puede controlar el sistema domótico ya sea EIB o Lonworks. De forma remota el control se realiza a través de Internet y órdenes IP.
- El mantenimiento se realiza de forma remota a través de Internet, es decir, cumple las especificaciones de pasarelas residenciales.

1.1.2 EUROCABLE, S.L.

- Control telefónico remoto de la vivienda.
- Aviso telefónico en caso de alarmas o incidencias técnicas.
- Control remoto y visualización de la vivienda a través de Internet.
- Control mediante mensajes cortos.

1.1.3 DOMOTICUS

- Las personas se pueden comunicar con la vivienda mediante pulsadores convencionales, botoneras domóticas, pantallas táctiles, mandos a distancia y con internet a través de teléfonos móviles, tabletas, smartTV's, ordenadores y también por reconocimiento de voz o por movimientos corporales entre otros.
- La casa se puede comunicar con las personas mediante los elementos domóticos de la instalación y también por Apps, correos electrónicos, mensajes y llamadas telefónicas.

1.1.4 INDOMO

- La casa se puede comunicar desde dentro fuera a través de cualquier Smartphone o teléfono IP.
- Aplicación móvil (TalkToKNX) para controlar por voz tu vivienda.

1.1.5 RESUMEN

	INTERNET		
	PLANTA	ORDENES	SEGURIDAD
DOMOTICA VIVA	X	X	CIFRADO
EUROCABLE		X	USUARIO/CONTRASEÑA
DOMOTICUS		X	USUARIO/CONTRASEÑA
INDOMO		X	USUARIO/CONTRASEÑA

Tabla 1. Tecnologías existentes, internet.

	TELEFONO			CONTROL
	AVISO	SMS	APP	VOZ
DOMOTICA VIVA				
EUROCABLE	X	X		
DOMOTICUS		X	X	X
INDOMO			X	X

Tabla 2. Tecnologías existentes, teléfono y voz.

1.1.6 VALORACIÓN

En el proyecto queremos reunir las siguientes características: seguridad y accesibilidad externa desde cualquier dispositivo, todo para crear un producto con las mismas competencias con respecto a las empresas actuales, por lo que ciframos el servidor y creamos una aplicación web adaptada a móviles, sin necesidad de envió de sms o control por voz.

1.2 ESTUDIO DE LOS DIFERENTES DISPOSITIVOS DE PROPÓSITO GENERAL

1.2.1 DEFINICIÓN DE LOS REQUISITOS

El nuevo dispositivo hardware debe disponer de unas especificaciones, firmware y controladores lo más libres posible, ser de pequeño tamaño, bajo coste, silencioso, y con unos consumos de energía reducidos. Además de esto, debe tener la capacidad de procesamiento necesaria para ejecutar el servidor web y los diferentes módulos y software que lo dotan de diferentes funcionalidades.

1.2.2 ALTERNATIVAS

Carambola

Figura 1. Carambola

1. Especificaciones:

• CPU: RT3050 a 2.4 Ghz.

• Memoria Flash: 8 MB.

• Memoria RAM: 32 MB.

• Red: Estándar Wireless 802.11 bgn, puerto para antena en el chip.

• Alimentación: 3.3v.

• Consumo energético: 1.5 W.

• Dimensiones: 35 x 45 mm.

2. Precio: 22 Euros aproximadamente.

Cubieboard

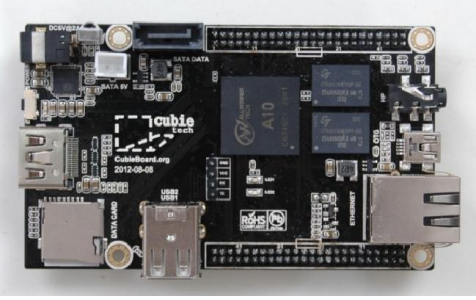

Figura 2. Cubieboard

1. Especificaciones:

• CPU: 1G ARM cortex-A8, NEON, VFPv3, 256KB caché L2.

• GPU: Mali400, OpenGL ES.

• Memoria: 512M/1GB DDR3 480MHz.

• Salidas vídeo: HDMI 1080p.

• Red: 10/100 Ethernet.

• Memoria Flash: 4Gb.

• Puertos USB: 2 puertos.

• Almacenamiento: 1 ranura micro SD, 1 puerto SATA, y 1 puerto IR.

2. Precio: 50 euros aproximadamente.

HackBerryA10

Figura 3. HackBerryA10

1. Especificaciones:

• CPU: 1.2 Ghz Allwinner A10 ARM Cortex A8

• GPU: Mali400 con aceleración 3D y decodificación de video por hardware.

• Memoria: 512 MB o 1 GB, reservando 100 MB aproximadamente para la GPU.

• Salida de video: HDMI 1080p, 3.5 mm AV compuesta, 3.5 mm por componentes Y/Pb/Pr

• Salida de audio: HDMI.

• Entrada de audio: jack 3.5 mm para micrófono.

• Puertos USB: 2 puertos.

• Almacenamiento interno: 4 GB NAND.

• Almacenamiento externo: Tarjetas SDHC de hasta 32 GB.

• Red: 10/100 Ethernet y Realtek 802.11n WiFi.

• Alimentación: Adaptador NEMA de 2 pines. Entrada AC100-240V-0.4A 50/60 . Salida DC5v.

2. Precio: 60-70 euros aproximadamente según memoria RAM.

Raspberry Pi

Figura 4. Raspberry Pi

1. Especificaciones:

• CPU: ARM1176JZF-S (ARM11) a 700 Mhz.

• GPU: Broadcom VideoCore IV, OpenGL ES 2.0, -2 y VC-1 (con licencia), 1080p30

H.264/MPEG-4 AVC.

• Memoria: 512 MB SDRAM compartidos con la GPU.

• Puertos USB: 2 puertos.

• Salidas vídeo: Conector RCA (PAL y NTSC), HDMI (rev1.3 y 1.4), Interfaz DSI para

panel LCD.

• Salidas Audio: 3.5 mm jack, HDMI.

• Almacenamiento: SD / MMC / ranura para SDIO.

• Red: 10/100 Ethernet (RJ45) via hub USB.

• Consumo energético: 700 mA, (3.5 W).

• Alimentación: 5V vía Micro USB o GPIO header.

• Dimensiones: 85.60mm × 53.98mm (3.370 × 2.125 inch).

2. Precio: 40 euros aproximadamente.

1.2.3 VALORACIÓN

De entre todas las opciones disponibles, se ha optado por escoger el dispositivo Raspberry Pi para el proceso de portabilidad de su sistema embebido. Las razones por las cuales se ha seleccionado este dispositivo son fundamentalmente las siguientes:

– Éxito del dispositivo, lo que conlleva una gran comunidad de usuarios y desarrolladores alrededor del mismo a nivel mundial.

– Aunque usa firmware y algunos controladores de código cerrado, las especificaciones y la mayor parte de los drivers son libres.

– Bajo coste del dispositivo.

– Sus prestaciones, que lo capacitan para desempeñar la función de controlador domótico.

– Reducido consumo y dimensiones.

1.3 MOTIVACIÓN DEL PROYECTO

En los últimos años el desarrollo de la domótica y la automatización del hogar han tenido un gran auge, y al paso del tiempo toma mayor importancia en la vida diaria de las personas, siempre buscando un camino más fácil de organizar su vida.

La principal motivación es contribuir al desarrollo de la domótica sin pagar un precio excesivo, en nuestro caso, poder monitorizar y controlar la vivienda de una manera segura desde cualquier sitio con conexión a internet.

BAJO COSTE

- Aprovechando la tecnología ya disponible.
- Uso de dispositivos de propósito general: Raspberry pi.

CONEXIÓN REMOTA SEGURA

- Servidor web cifrado (https).

Motivación común a instalaciones ya existentes, confort y ahorro energético.

CONFORT

- Control remoto y visualización de la vivienda a través de cualquier dispositivo con acceso a internet: móvil, PC, etc.
- Control de temperatura: por presencia, regulación de intensidad, etc

AHORRO ENERGÉTICO

- Gestión de Climatización (calefacción/aire acondicionado) por programación horaria o control de temperatura por zonas.
- Gestión de Consumo de calefactores: programación para funcionamiento en franjas horarias más económicas.

1.4 OBJETIVOS

Creación de aplicación web para el posterior control de diferentes elementos de la casa.

1. Esquema del plano de la vivienda con monitorización y control.
2. Registro histórico de datos de temperatura.

1.5 METODOLOGÍA / SOLUCIÓN DESARROLLADA

El método para abordar el proyecto se divide en cuatro partes principales:

1.- Configuración Raspberry Pi I y II

2.- Instalación del servidor

3.- Configuración del router

4.- Creación de base de datos

5.- Desarrollo de la programación I y II

1.5.1 CONFIGURACIÓN LA RASPBERRY PI I

Una vez instalado el sistema operativo Raspbian en la Raspberry, la primera configuración debe ser SSH, para poder acceder a la Raspberry Pi de forma remota, es decir, poder realizar cualquier modificación de la Raspberry desde otro dispositivo diferente sin necesidad de tener conectado un TV o monitor.

Para tal fin es necesario establecer una ip estática, modificando la configuración de red, accediendo al archivo con la siguiente ruta:

```
/etc/network/interfaces
```

y cambiando la configuración de conexión externa (wifi)por:

```
iface lo inet loopback
iface eth0 inet dhcp
allow-hotplug wlan0
iface wlan0 inet manual    //Conexión manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf

iface default inet static  //Parámetros de configuración
```

address 192.168.1.137

gateway 192.168.1.1

netmask 255.255.255.0

1.5.2 INSTALACIÓN DEL SERVIDOR

Una vez configurada la Raspberry Pi empezaremos a instalar el servidor Web, nuestro servidor web dispondrá de los elementos básicos:

- Un servidor web o software que permita a nuestro ordenador alojar páginas web dinámicas. En nuestro caso utilizaremos Apache, un software de código abierto y libre que funciona en cualquier sistema operativo.
- El entorno PHP compuesto por un conjunto de programas que nos permitirán solicitar programas escritos en este lenguaje de programación desde un servidor web, que posteriormente se mostrarán en el navegador en HTML.
- Un sistema de gestión de bases de datos con las que los programas en PHP interactuarán. En nuestro caso utilizaremos MySQL un sistema de bases de datos relacional que se distribuye bajo licencia GNU GPL para proyectos que se distribuyan bajo esa misma licencia.
- Un administrador de MySQL a través de páginas web, utilizando Internet, que permite crear y manejar bases de datos con una interfaz más sencilla, PhpMyAdmin.

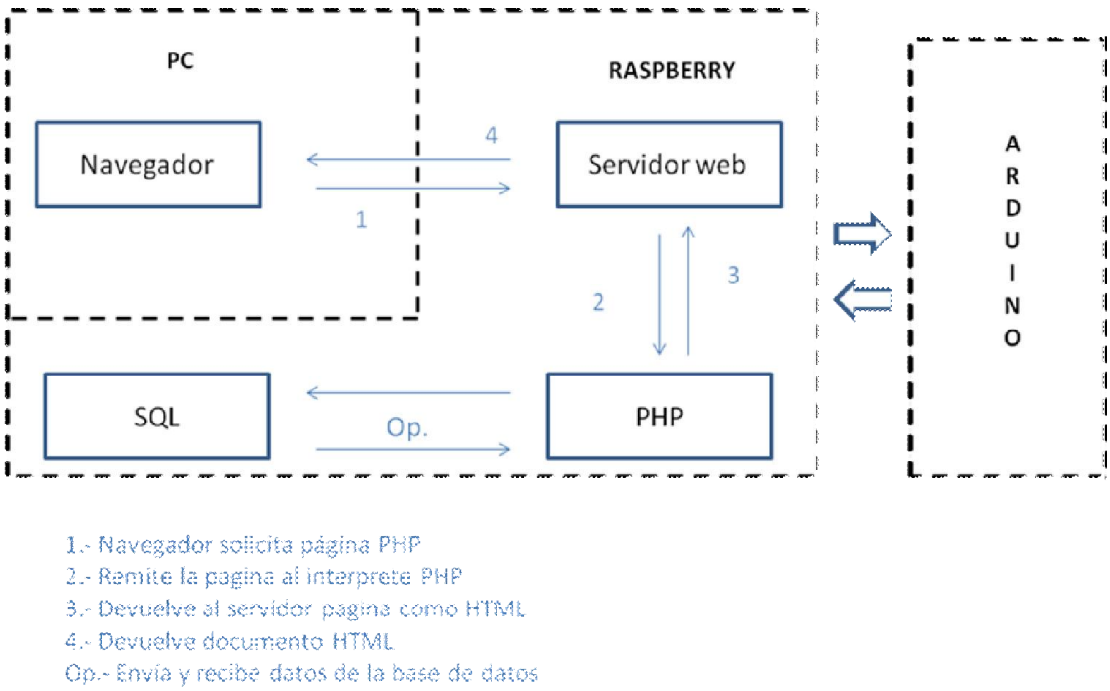

Figura 5. Funcionamiento servidor.

El proceso comienza, antes de la instalación, con la actualización de programas y repositorios

```
sudo groupadd www-data
sudo usermod -a -G www-data www-data
```

Instalamos Apache

```
sudo apt-get install apache2
```

Instalamos PHP para poder crear contenido dinámico en nuestra web

```
sudo apt-get install php5
```

junto con unos paquetes que nos podrían hacer falta

```
sudo apt-get install libapache2-mod-php5 libapache2-mod-perl2 php5 php5-cli php5-common php5-curl php5-dev php5-gd php5-imap php5-ldap php5-mhash php5-mysql php5-odbc
```

Reiniciamos la Raspberry y procedemos a la instalación de la base de datos MySQL.

```
sudo apt-get install mysql-server mysql-client php5-mysql
```

Instalación de PhpMyAdmin.

```
sudo apt-get install libapache2-mod-auth-mysql php5-mysql phpmyadmin
```

1.5.3 CONFIGURACIÓN LA RASPBERRY PI II

Permisos

Una vez instalado el servidor otorgamos permisos:

Los directorios típicamente utilizados por un servidor web en Linux se sitúan en la carpeta, /var/www, y el usuario típico para dicho entorno suele ser www-data. Ahora vamos a crear el grupo y usuario estándar para nuestro servidor, a la par que otorgamos los permisos pertinentes y añadimos a nuestro usuario por defecto (pi) al grupo comentado. De esta forma no será preciso que el usuario root (su) sea siempre el que pueda operar en la ya carpeta mencionada, /var/www.

Cambiamos el usuario y grupo al directorio

```
sudo chown www-data:www-data /var/www/
```

Damos los permisos a la carpeta www/

```
sudo chmod 775 /var/www/
```

Añadimos el usuario pi al grupo www-data

```
sudo usermod -a -G www-data pi
```

Reiniciamos el servidor.

```
sudo /etc/init.d/apache2 restart
```

Seguridad

Puesto que una de las principales cuestiones que se planteaban, era el problema de la seguridad en internet, se opto por la decisión de cifrar el servidor como medida de mayor garantía ante dicha inseguridad.

Incorporamos los certificados de cifrado

```
sudo mkdir /etc/apache2/ssl
```

Generamos los certificados para 3 años (1095 dias)

```
sudo openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -out
/etc/apache2/ssl/server.crt -keyout /etc/apache2/ssl/server.key
```

Instalamos el mod SSL para Apache

```
sudo a2enmod ssl
```

Configuramos el sitio predeterminado SSL y lo habilitamos

```
sudo ln -s /etc/apache2/sites-available/default-ssl /etc/apache2/sites-enabled/000-
default-ssl
sudo nano /etc/apache2/sites-enabled/000-default-ssl
```

y cambiamos las dos líneas en relación con el certificado SSL

```
SSLCertificateFile    /etc/apache2/ssl/server.crt
SSLCertificateKeyFile /etc/apache2/ssl/server.key
```

y reiniciamos el servidor

```
sudo /etc/init.d/apache2 restart
```

Figura 6. Ejemplo del cifrado

Acceso externo

Para acceder desde el exterior debemos conocer nuestra ip, dado que tenemos una configuración ip dinámica, cada vez que reiniciamos está cambia, debemos conocer nuestra ip para la conexión, pero no nos planteamos tener un ip fija, dado que cuesta dinero.

Vamos a utilizar el servicio de DNS dinámica de No-IP, que permite identificar tu PC con un nombre de dominio fácil de recordar, como en nuestro caso, domo.no-ip.biz en lugar de recordar la ip sin importar si está cambia.

Descargamos el paquete, lo descomprimimos y lo instalamos

```
wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz
tar -zxvf noip-duc-linux.tar.gz
cd noip-2.1.9-1/
make
sudo make install
```

Configuramos no-ip para que arranque automáticamente con el encendido de la Raspberry, para ello modificamos el siguiente archivo con ruta: /etc/init.d/noip2

y copiamos el siguiente texto:

```
#! /bin/bash
### BEGIN INIT INFO
# Provides: blabla
# Required-Start: $syslog
# Required-Stop: $syslog
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: blabla
# Description:
#
### END INIT INFO
sudo /usr/local/bin/noip2
```

Guardamos. Le damos permisos de ejecución y lo ponemos en la cola de ejecución

```
sudo chmod +x /etc/init.d/noip2
sudo update-rc.d noip2 defaults
```

1.5.4 CONFIGURACIÓN DEL ROUTER

Para terminar de configurar el acceso externo hay que abrir ciertos puertos del router, ya que un servidor es una aplicación que está diseñada para enviar datos hacia aquellos usuarios que se los solicitan, esta aplicación sí precisa del mapeo de varios puertos en caso de ejecutarse en sistemas que se conectan a Internet a través de dispositivos NAT

Los puertos a abrir son:

PUERTOS	SERVICIO
22	SSH
21	FTP
80	HTTP
5900-5910	VNC

Tabla 3. Puertos abiertos router.

En mi caso directamente aplique la configuración DMZ(zona desmilitarizada) disponible en mi router, sin abrir los puertos anteriormente descritos.

El funcionamiento de esta herramienta consiste en determinar una zona que sería una puerta desde nuestro router hacia internet sin ninguna barrera de por medio, es decir, una salida a internet con todos los puertos abiertos. Dicha Zona o salida se aplica a un solo equipo de nuestra red doméstica, en nuestro caso la dirección ip interna de la Raspberry Pi.

DMZ:

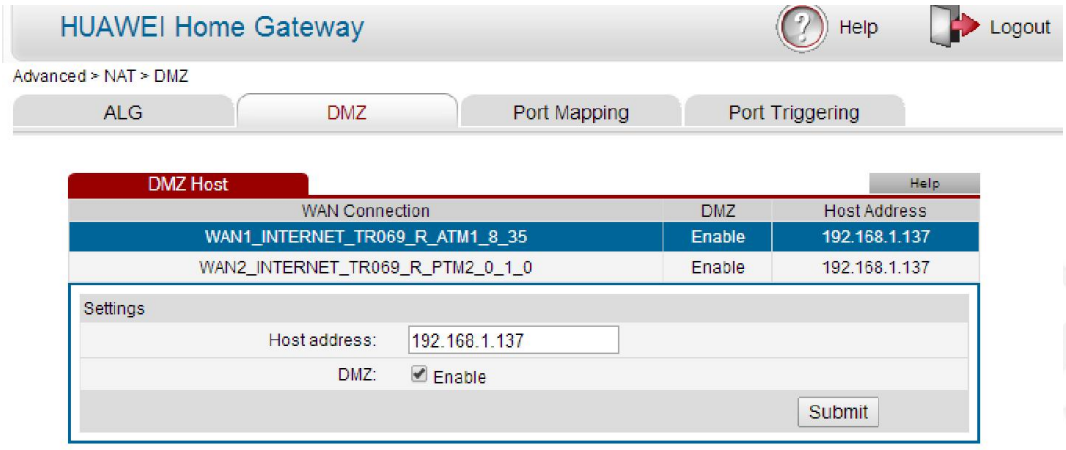

Figura 7. Configuración DMZ del router.

1.5.5 DESARROLLAR LA PROGRAMACIÓN I

En este punto recorremos todo el proceso necesario para desarrollar la aplicación web. Para ello en un primer instante se define la arquitectura de la pagina web, es decir la forma de presentar el contenido, para ello se descargo una plantilla gratuita ofrecida por la pagina web estudio365, elección del modelo "Response",

debido a que es una plantilla perfectamente adaptada a móviles, con este estilo de diseño, los elementos que componen la web se reorganizan de forma adecuada en cualquier tipo de dispositivo móvil o de sobremesa.

Hoy en día existen más personas navegando por Internet a través de dispositivos móviles, como portátiles, tabletas o teléfonos móviles que a través de ordenadores de sobremesa, por tanto contar con una página web optimizada para dispositivos móviles ayuda a mejorar la experiencia de navegación del usuario así como su acceso.

Con el diseño adaptado y estructurado de nuestra página web empezamos a utilizar la herramienta CANVAS de HTML5 para crear un primer diseño de la planta de la vivienda.

Comenzamos situando dentro del cuerpo de la página(<body>) la etiqueta CANVAS. Esta etiqueta forma parte del estándar del HTML 5.Defimos el tamaño y con el atributo id, un nombre único para luego referirnos a este desde Javascript.

Definición de zona CANVAS,

```
<canvas id="micanvas" width="500" height="400">Recicla tu browser, POR FAVOR!!</canvas>
```

Pintamos en un canvas mediante Javascript, inicialmente el canvas está en blanco y cuando queremos pintar sobre él tenemos que acceder al contexto de renderizado del canvas, sobre el que podremos invocar distintos métodos para acceder a las funciones del dibujo.

Primero con el método getElementById() obtengo el elemento de la página que se pasa como parámetro, que es el canvas. Luego accedo al contexto 2D del canvas, que es el que tiene varios métodos que sirven para dibujar en el lienzo. Por último ejecuto los métodos para pintar elementos en el lienzo, es decir la planta de la casa con los elementos a monitorizar y controlar.

Cargar elemento CANVAS, para más detalle ver archivo planta.php en sección código

```
function cargaContextoCanvas(idCanvas){
```

```
            //Recibimos el elemento canvas
            var elemento = document.getElementById(idCanvas);
            if(elemento && elemento.getContext){
                //Accedo al contexto de '2d' de este canvas
                var contexto = elemento.getContext('2d');
                if(contexto){


                //--PAREDES Y PUERTAS
                ctx.beginPath();
                //Desplazamiento
                ctx.moveTo(40,20);
                //Dibujar segmento
                ctx.lineTo(40,175);
                ctx.lineTo(150,175);
                //Desplazamiento
                ctx.moveTo(40,20);
                ctx.lineTo(190,20);
                ctx.lineTo(190,175);
                ctx.moveTo(177.5,175);
                ctx.lineTo(202.5,175);
                ctx.lineTo(202.5,135);
                ctx.moveTo(177.5,175);
                ctx.lineTo(177.5,135);
                //.....código completo en planta.php, sección
                del documento: código

                    return contexto;
                }
            }
            return FALSE;
        }
```

Esta vivienda cuenta con la representación de radiadores y una sencilla función que representa su funcionamiento.

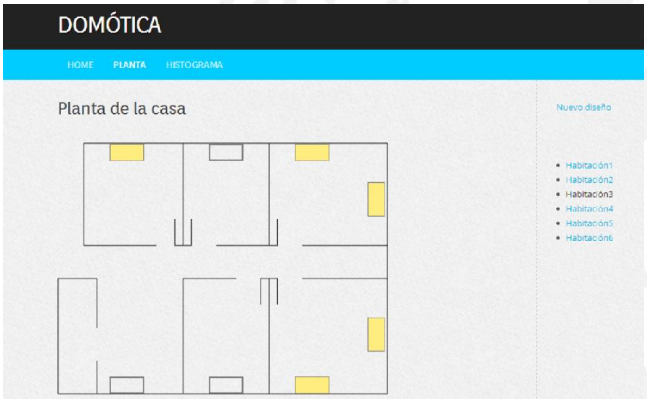

Figura 8. Representación de la planta de la vivienda.

Ejemplo funcionamiento, radiador 1

```
window.onload = function()
	{
	//Recibimos el elemento canvas
	var ctx = cargaContextoCanvas('micanvas');
	if(ctx)
	{
	ctx.strokeRect( 80, 22, 50,24); //Bordes negros radiador1
	//Coloreado inicial
	if(original == 0){
		//Radiador1
		if(sit1==1){
			ctx.fillStyle = '#ffee7f'; //Colorea de amarillo
			ctx.fillRect(80,22,50,24);
		}
		else if(sit1==0){
			ctx.fillStyle = 'white';	//Colorea de blanco
			ctx.fillRect(80,22,50,24);
		}
		original = 1;
	}
}
```

Habiendo hecho un primer contacto con respecto al primer objetivo, el segundo objetivo, el histograma, era el siguiente paso a realizar. Para la representación de la temperaturas se valoraron diferentes opciones, entre las que destacaba la resolución utilizando el ya bien conocido Canvas junto con Javascript y jQuery, frente a la utilización de una aplicación de Google denominada Google Charts, dada la facilidad de implementación, así como una mejor experiencia interactiva con el usuario, se opto por esta segunda opción.

Para hacer uso de esta herramienta hay que cargar los repositorios de la aplicación de Google y definir la función en Javascript que dibujará el gráfico, dicho código en la cabeza (<head>) de la página.

```
<!--JAVA. REPOSITORIO APLICACION GOOGLE-->
    <script type="text/javascript" src="https:
//www.google.com/jsapi"></script>
    <!--INFORMACIÓN HISTOGRAMA HABITACIÓN 1-->
    <script type="text/javascript">

google.load("visualization", "1", {packages:["corechart"]});
google.setOnLoadCallback(drawChart);
function drawChart() {
	var data = google.visualization.arrayToDataTable([
```

```
            ['Hour','Ayer','Hoy'],
            ['15:00',2,21],
            ['16:00',2,20],
            ['17:00',2,20],
            ['18:00',2,19],
            ['19:00',2,19],
            ['20:00',2,20],
            ['21:00',2,20],
            ['22:00',2,20]
          ]);

    var options = {
      title: 'Registro de temperaturas', //Titulo de la gráfica
      curveType: 'function', //Con líneas curvas
    };
    //Gráfica de tipo línea
    var chart = new
google.visualization.LineChart(document.getElementById('chart_div'
)); //id=char_div
       chart.draw(data, options);
    }
```

Para obtener el dibujo de la gráfica cargamos dentro del cuerpo (<body>), el elemento anterior definiendo el tamaño correspondiente.

```
<div id="chart_div" style="width: 600px; height:
200px;"></div>//Definicion del tamaño
```

Sabiendo cómo crear un gráfico de temperaturas para cada habitación, es momento de representar todas las gráficas en una misma página, se opto por una representación oculta nada más cargar la página que mostrase u ocultase cada una de las gráficas al hacer click en la opción correspondiente a cada habitación, de esta manera conseguimos una visualización más clara y accesible al usuario, ver figura 10.

La solución más sencilla es el uso de iframe, al hacer click se muestra la página correspondiente al archivo con la ubicación definida en el atributo src, dicha página contiene la gráfica correspondiente a la habitación, no olvidar definir diferentes identificadores correspondientes a las diferentes habitaciones con atributo id.

```
//Cada iframe se muestra y se oculta cada vez que se clickea, el
tamaño es todo el ancho con 250 de alto
 <a class="h_menu" onclick=" var ifreim1 =
document.getElementById('ifreim1'); ifreim1.style.display =
ifreim1.style.display == 'none' ? 'block' : 'none';
"/>Habitación1</a>
//Ruta del archivo histograma/h1 con el identificador ifreim1
<iframe src="histograma/h1.html" id="ifreim1" style="display:none"
width=100% height=250 display=none></iframe>
```

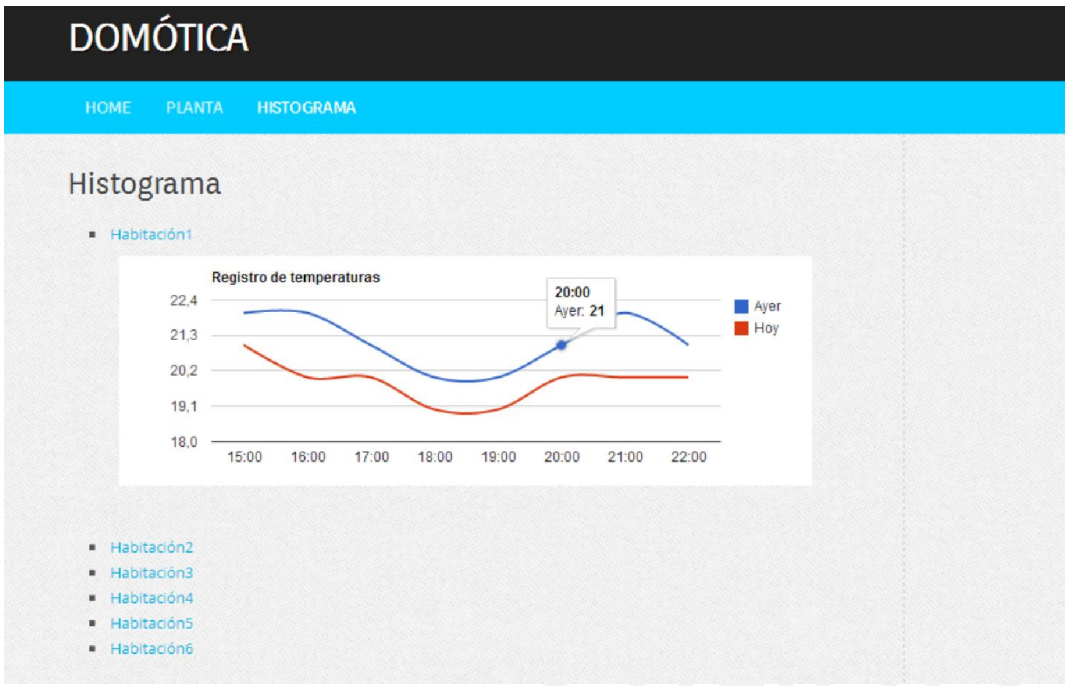

Figura 9. Histograma de temperatura.

Una vez centrados los dos objetivos principales, es momento de dinamizar los contenidos, es de vital importancia que lo mostrado en dicha aplicación este sincronizado con la situación actual de la vivienda, más en concreto con la situación actual de los dispositivos de la casa. La opción más sencilla es utilizar una base de datos con la situación de la vivienda, para extraer los datos de dicha base de datos debemos utilizar el lenguaje de programación PHP, es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Puesto que fue uno de los

primeros programas del lado del servidor que se incorporaban directamente en el documento HTML, se postula como la mejor opción.

1.5.6 CREACIÓN DE LA BASE DE DATOS

Una base de datos es la herramienta para guardar la información de la situación actual de la casa, así como también puede ser usada para guardar el registro histórico de temperaturas que se produce en cada habitación. La opción mejor considerada fue instalar el gestor de bases de datos MySQL en la Raspberry pi, por la compatibilidad con sistemas de código abierto y una gran comunidad que aportan gran información, se instaló con una licencia GNU GPL.

Con los programas necesarios instalados se procedió a la creación de una base de datos denominada *TEMPERATURA*, que contiene siete tablas *RADIADORES, HABITACIÓN1, HABITACIÓN2, HABITACIÓN3, HABITACION4, HABITACION5, HABITACION6* que contienen la información con la situación de los radiadores (encendido/apagado) y con el registro de temperaturas de cada habitación, respectivamente. La codificación de ambas tablas es UTF-8 (formato de codificación de caracteres Unicode e ISO 10646 utilizando símbolos de longitud variable), la misma codificación establecida en el servidor y en las páginas.

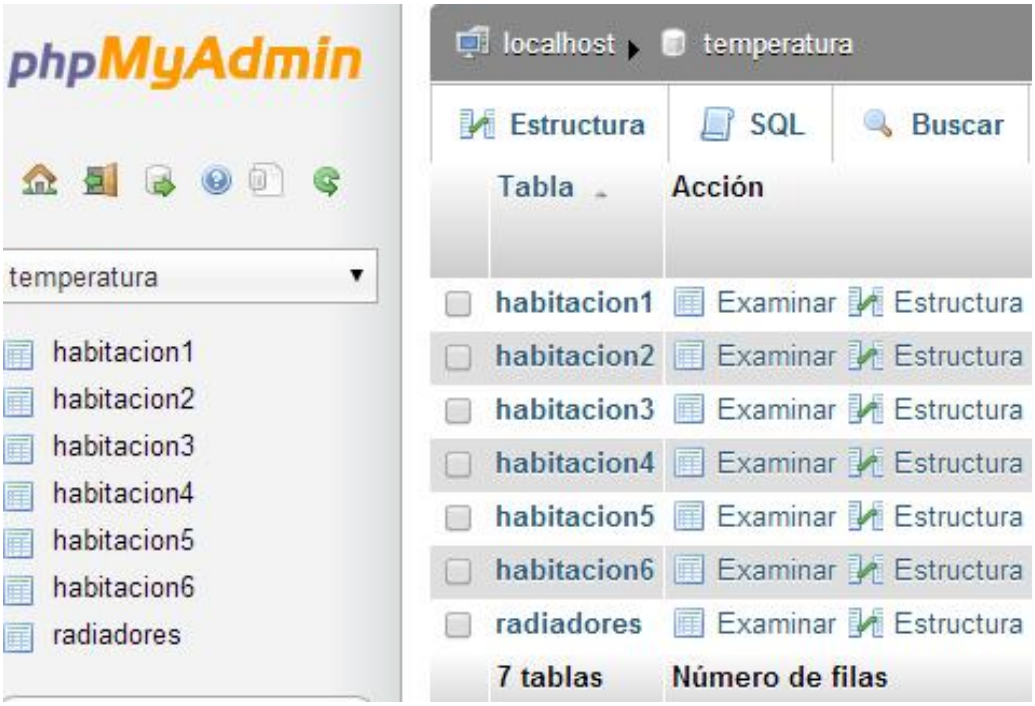

Figura 10. PhpMyAdmin

La tabla RADIADORES contiene 3 campos de información:

- id: identificación, tipo entero, auto incrementable.
- nombre: información de la habitación, tipo cadena de caracteres.
- estado: información del estado, tipo numérico: 1 encendido ó 0 apagado.

#	Columna	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
1	id	int(11)			No	*Ninguna*	AUTO_INCREMENT
2	Nombre	varchar(20)	utf8_spanish_ci		No	*Ninguna*	
3	Estado	int(11)			No	*Ninguna*	

Figura 11. Variables tabla Radiadores.

Las tablas HABITACION_ contiene 3 campos de información útiles:

- id: identificación, tipo entero, auto incrementable.
- TempHoy: información temperatura hoy, tipo entero.
- TempAyer: información temperatura ayer, tipo entero.

#	Columna	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
1	id	int(11)			No	Ninguna	AUTO_INCREMENT
2	TempHoy	int(11)			No	Ninguna	
3	TempAyer	int(11)			No	Ninguna	

Figura 12. Variables tabla habitacion1

1.5.7 DESARROLLO DE LA PROGRAMACIÓN II

La página está desarrollada en HTML, es estática, por lo que hay que desarrollar la programación PHP para determinarla como una página dinámica, para ello hay que ser capaz de realizar consultas, de mostrar y actualizar los datos de las tablas de la base de datos.

En un primer momento hay que realizar la conexión, el código es:

```
<?php
//Declaración de variables PHP y de acceso a BBDD
$base_datos = "temperatura";
$tabla2 = "radiadores";
$direccion_bd = "localhost";
$usu_bd  = "root";
$pass_bd = "-------";
// Conectamos con BBDD
$conexion = mysql_connect ($direccion_bd,$usu_bd,$pass_bd) or die
("No es posible acceder a la BBDD");
?>
```

La primera acción a realizar es una consulta y su consecuente representación.

```
<?php
include("../conexion.php");

//Solicitamos la consulta.
$consulta1="SELECT * FROM habitacion1";

//Seleccionamos la tabla.
$resultado1=mysql_db_query ($base_datos,$consulta1,$conexion);

//Seleccionamos el valor de la 1ªfila y de la columna TempHoy.
$h1=mysql_result($resultado1,0,"TempHoy");
```

```
//Mostramos el valor.
echo $h1;

//Cerramos conexión.
mysql_close($conexion);
?>
```

Con el código anterior podemos utilizar los valores de la base de datos para la representación de temperaturas de cada habitación así la representación de estado de los radiadores, ver figura 14 y 15.

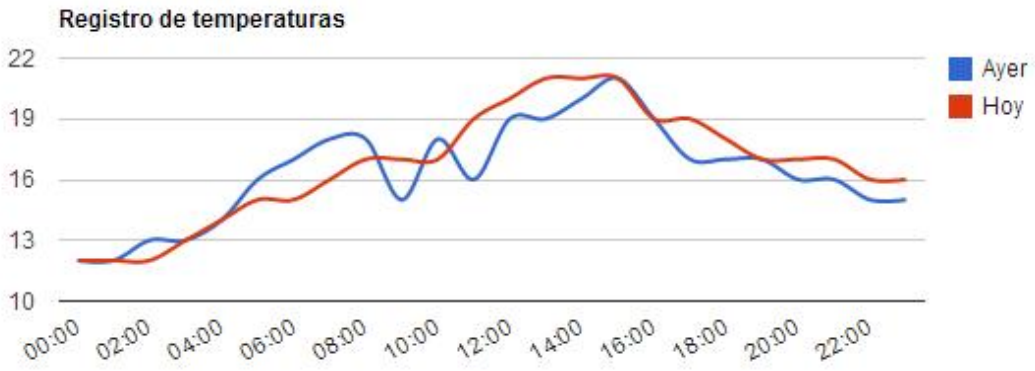

Figura 13. Representación de temperaturas con datos BBD

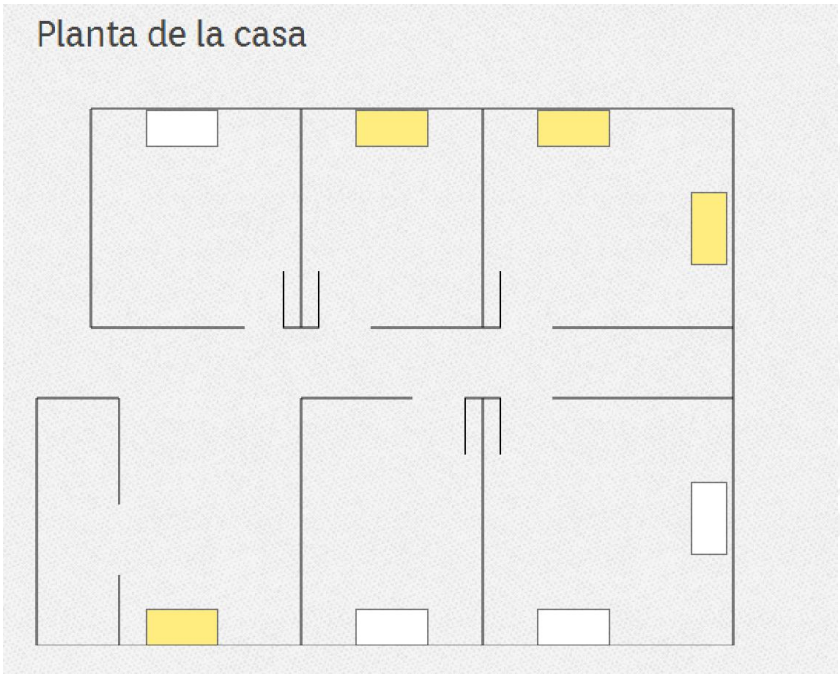

Figura 14. Representación de habitación con datos BBD

El siguiente punto a abordar es la actualización de los radiadores, el funcionamiento que se quiere representar es similar al de un interruptor, al pulsar se encenderá o se apagará en función de su estado, cada interruptor llevara el nombre de la habitación correspondiente, ver figura 15.

Se nos presenta un problema pues no es posible llamar desde HTML a ninguna función PHP, puesto que el primero tiene su ámbito en el cliente y el segundo lo tiene en el Servidor.

La solución para representar el interruptor, consiste en utilizar un formulario HTML, que es una de las formas "naturales" de comunicación del cliente con el servidor y viceversa, además de para así poder representar un interruptor con <button>. El atributo "action" del formulario HTML se utiliza para indicar quién responderá del proceso del formulario, quién recibirá las variables y valores que el usuario puede introducir a través del formulario. Definimos método de envió como POST, método propio de los formularios.

```
<form action="actualizaciones/h1.php" method="post">
<button class="mult_submit" type="submit" name="submit_mult"
value="Modificar" title="Radiador1">Habitación1</button>
</form>
```

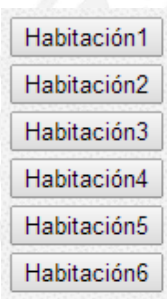

Figura 15. Representación de interruptores

La página a la que redirige es *actualizacion/h1*, que contiene código para actualizar la base de datos y código para re direccionar a *../planta*, para ver los cambios realizados.

El código PHP que actualiza la base de datos

```php
<?php
//Solo código php
//Incluimos conexión a base de datos.
include("../conexion.php");

//Seleccionamos la base de datos.
mysql_select_db("temperatura",$conexion);

//Actualizamos la el estado del radiador 1.
$ssql = "UPDATE radiadores SET Estado=1 WHERE id=1";

//Cerramos conexión.
mysql_close($conexion);
?>
```

El código que re direcciona a planta.php tras 5 segundos.

```
//Redireccionar a ...planta.php en 5 segundos.
<meta http-equiv="Refresh" content="5; url=../planta.php" />
```

1.6 RECURSOS / HERRAMIENTAS EMPLEADAS

1.6.1 RASPBERRY PI

Raspberry Pi es un ordenador de placa reducida o (placa única) (SBC) de bajo costo, desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.

El diseño incluye un System-on-a-chip Broadcom BCM2835, que contiene un procesador central (CPU) ARM1176JZF-S a 700 MHz , un procesador gráfico (GPU) VideoCore IV, y 512 MiB de memoria RAM . El diseño usa una tarjeta SD para el almacenamiento permanente.

1.6.2 RASPBIAN

Raspbian es una distribución del sistema operativo GNU/Linux y por lo tanto libre basado en Debian Wheezy (Debian 7.0) para la placa computadora (SBC) Raspberry Pi, orientado a la enseñanza de informática. El lanzamiento inicial fue en junio de 2012.

1.6.3 APACHE

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd, pero más tarde fue reescrito por completo.

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

1.6.4 PHP

PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante.

1.6.5 CODECADEMY

Codecademy es una plataforma interactiva en línea que ofrece clases gratuitas de codificación en lenguajes de programación como Python, PHP, JavaScript, y Ruby, así como lenguajes de marcado incluyendo HTML y CSS.

1.6.6 CANVAS

El elemento canvas es parte de HTML5 y permite la renderización scriptable dinámica de formas 2D e imágenes de mapa de bits. Se trata de un modelo de procedimiento de nivel bajo, el que actualiza un mapa de bits y no tiene una gráfica de escena integrada.

El Canvas consiste en una región dibujable definida en el código HTML con atributos de altura y ancho. El código JavaScript puede acceder a la zona a través de un conjunto completo de funciones similares a las de otras API's comunes de dibujo 2D, permitiendo así que los gráficos sean generados dinámicamente. Algunos de los usos previstos incluyen construcción de gráficos, animaciones, juegos, y la composición de imágenes.

1.6.7 GOOGLE CHARTS API

La API de gráficos de Google es una herramienta que permite a los usuarios crear fácilmente un gráfico a partir de algunos datos e incrustarlo en una página web. Google crea una imagen PNG de un gráfico de los datos y parámetros de formato en una solicitud HTTP. Se admiten muchos tipos de gráficos como por ejemplo: líneas, barras, circulares y de gráficos de radar, así como los diagramas de Venn, diagramas de dispersión, mapas.

1.6.8 MYSQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario, servicio usado por muchos sitios web grandes y populares, como Wikipedia, Google.

MySQL AB, desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009, desarrolla MySQL como software libre en un esquema de licenciamiento dual.

MySQL es muy utilizado en aplicaciones web, como Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL.

1.6.9 PHPMYADMIN

PhpMyAdmin es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet. Actualmente puede crear y eliminar bases de datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 62 idiomas. Se encuentra disponible bajo la licencia GPL.

Esta herramienta corre en máquinas con Servidores Web y Soporte de PHP y MySQL, la tecnología utilizada ha ido variando durante su desarrollo, actualmente está disponible en 69 idiomas, y se encuentra mantenido por el Proyecto PhpMyAdmin bajo responsabilidad de Olivier Müller, Marc Delisle, Alexander M. Turek, Michal Čihař y Garvin Hicking.

Capítulo 2 CONCLUSIONES

El desarrollo de la aplicación que se presenta en este proyecto surge de la necesidad de control y monitorización de los elementos domóticos de la casa.

Para abordar el diseño de la aplicación propuesta, se han estudiado las posibles soluciones de software, enumerando y destacando en la memoria de este documento aquellas que, por sus ventajas desde el punto de vista de la arquitectura, se han elegido para realizar la aplicación. Entre los motivos enumerados destaca, el bajo coste para la implementación del sistema domótico así como el confort de controlar y manejar de manera segura los diferentes elementos eléctricos que componen la vivienda.

Se ha conseguido una aplicación eficaz y eficiente, a la par que fiable y estable, con una curva de aprendizaje mínima y con la funcionalidad más utilizada a la vista, evitando complejos procedimientos para su uso.

Capítulo 3 FUTUROS DESARROLLOS

3.1 CONEXIÓN SPI

3.1.1 INTRODUCCIÓN

El Bus SPI (Serial Peripheral Interface) es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos. El bus de interfaz de periféricos serie o bus SPI es un estándar para controlar casi cualquier dispositivo electrónico digital que acepte un flujo de bits serie regulado por un reloj.

Nuestra intención es usar la conexión SPI para comunicar la Raspberry Pi con el Arduino correspondiente al otro proyecto, más en concreto enlazar la información de nuestra base de datos con la información guardada en el Arduino.

Antes de empezar con el código vamos a explicar algunos detalles del SPI de la Raspberry.

1.- La Raspberry tiene dos puertos SPI el 0 y el 1, representados en la Raspberry Pi como dos Chips enable que en tienen el nombre de CE0 y CE1.

2.- La velocidad a la que puede comunicarse va desde 500 KHz y 32 MHz, siendo una velocidad óptima 1MHz .

Pines usados:

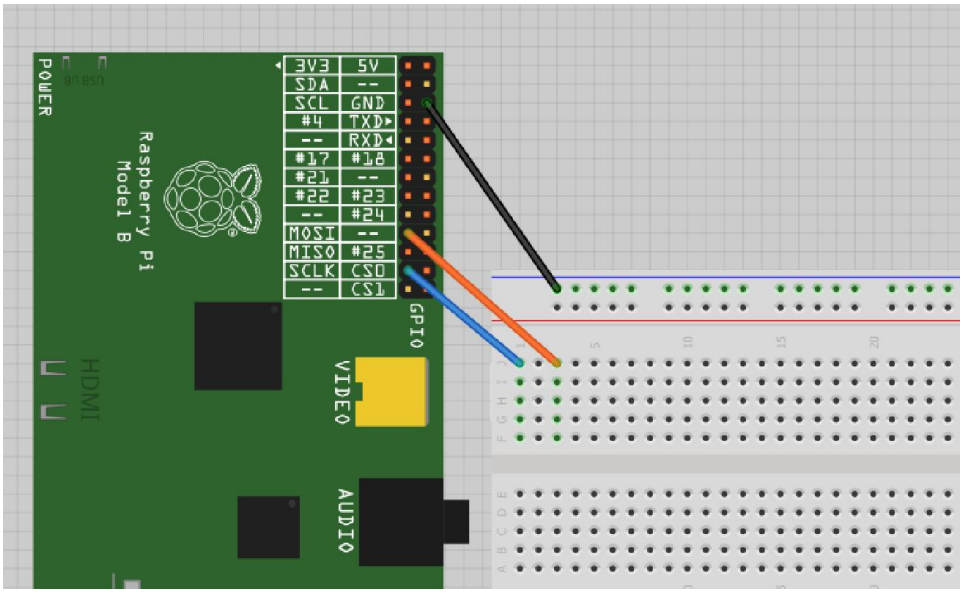

Figura 16. Conexión SPI

3.1.2 INSTALACIÓN Y USO DE LIBRERÍA WIRINGPI

Instalación

Cargar drivers SPI

gpio load spi

Uso

Básicamente hay dos únicas funciones en esta librería, una para configurar y otra para leer/escribir, ambas se utilizaran desde un archivo C.

int wiringPiSPISetup (int channel, int speed) ; :Función para configurar el spi. En channel ponemos 1 o 0 según el puerto que queramos usar y en speed ponemos

la velocidad en Herzios de nuestra comunicación SPI. Si la función nos devuelve un -1 quiere decir que no se ha iniciado bien y que tenemos un error.

int wiringPiSPIDataRW (int channel, unsigned char *data, int len) ; : Por ultimo tenemos la función de escritura/lectura, esta tiene 3 parámetros El canal , el buffer de datos y el tamaño de los datos a enviar. Dado que esta función lee y escribe al mismo tiempo, lo que pongamos en data sera sobrescrito por lo que se lea del bus.

3.1.3 CONFIGURACIÓN DEL ACCESO A MYSQL DESDE C

Instalación

Biblioteca C en MySQL

sudo apt-get instalar libmysqlclient-dev

Corrección del archivo MAKE para poder acceder a MySQL desde código C.

Línea 8: *LIBS_1 = `mysql_config - libs`*
Línea 10: *CFLAGS_1 =-Wall-c $ (DEBUG) `mysql_config - cflags`*

3.1.4 COMUNICACIÓN CON MYSQL

Desde archivos escrito en C++.

Conexión

```
#include <mysql/mysql.h>
# Define SERVER "localhost"
# Define USUARIO "usuario"
# Define PASSWORD "top_secret_password"
# Define DATABASE "database"

//********** Conectar con base de datos **********

void mysql_connect (void)
```

```
{
    //initialize MYSQL object for connections
    mysql1 = mysql_init(NULL);

    if(mysql1 == NULL)
    {
        fprintf(stderr, "%s\n", mysql_error(mysql1));
        return;
    }

    //Connect to the database
    if(mysql_real_connect(mysql1, "localhost", DATABASE_USERNAME,
DATABASE_PASSWORD, DATABASE_NAME, 0, NULL, 0) == NULL)
    {
        fprintf(stderr, "%s\n", mysql_error(mysql1));
    }
    else
    {
        printf("Database connection successful.\n");
    }
}
```

Desconexión

```
void mysql_disconnect (void)
{
    mysql_close(mysql1);
    printf( "Disconnected from database.\n");
}
```

Actualizando la base de datos.

```
void mysql_write_something (void)
{
   //vector times;    //a vector of alarm times

    if(mysql1 != NULL)
    {
        //Retrieve all data from alarm_times
        if (mysql_query(mysql1, "INSERT INTO settings (    \
                    id,    \
                    value_string    \
                ) VALUES (    \
                    99,    \
                    'Hello'    \
                ) \
                ON DUPLICATE KEY UPDATE    \
                    id = 99,    \
                    value_string = 'Hellow'    \
                "))

        {
            fprintf(stderr, "%s\n", mysql_error(mysql1));
```

```
            return;
        }
    }
}
```

Consultando de la base de datos.

```
    mysql_connect();

    if (mysql1 != NULL)
    {
        if (!mysql_query(mysql1, "SELECT value_int, value_string
FROM settings WHERE id = 8"))
        {
            MYSQL_RES *result = mysql_store_result(mysql1);
            if (result != NULL)
            {
                //Get the number of columns
                int num_rows = mysql_num_rows(result);
                int num_fields = mysql_num_fields(result);

                MYSQL_ROW row;          //An array of strings
                while( (row = mysql_fetch_row(result)) )
                {
                    if(num_fields >= 2)
                    {
                        char *value_int = row[0];
                        char *value_string = row[1];

                        printf( "Got value %s\n", value_string);
                    }
                }
                mysql_free_result(result);
            }
        }
    }
    mysql_disconnect();
```

Capítulo 4 BIBLIOGRAFÍA

[RIVE13] Documento: Desarrollo de un sistema demótico básico usando Raspberry Pi, PHP y Python. Autor: Jefferson Rivera Patiño.

[DIEG13] Libro: El gran libro de HTML5, CSS3 y Javascript. Autor: Juan Diego Gauchat. Editorial: marcombo

[SANC13] Documento: Administración de Redes y Sistemas Operativos en Entornos de Softwate Libre. Autor: David Sánchez Herrero. Tutor: Oriol Palenzuela i Rosés

[1] Machuca, C. A. (s.f.). Obtenido de http://www.methesis.fcs.ucr.ac.cr/blogs/rumana/wp-content/uploads/2010/09/articulo2.pdf

[2] *Seguridad informatica*. (s.f.). Obtenido de http://seguridadinformaticaufps.wikispaces.com/file/view/ESTADO+DEL+ARTE+SI.pdf

[3] http://www.codecademy.com/es

[4] https://developers.google.com/chart

[5] http://www.mysql.com

[6] http://www.phpmyadmin.net/home_page/index.php

[7] http://www.electroensaimada.com

[8] http://www.desarrolloweb.com/
http://piro.wordpress.com/2012/11/22/raspberry-pi-static-wifi-ip

[9] http://www.estudio365.com.ar/25-plantillas-web-gratis-en-html5-y-css3-para-diseno-web-adaptado-a-movil

[10] http://www.forosdelweb.com/f13/consulta-como-esconder-iframe-con-javascript-346397

[11] http://www.php.net/manual/es/book.mysql.php

[12] http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=612:consultas-basicas-a-mysql-en-php-mysql-connect-close-selectdb-query-y-result-ejemplos-cu00838b&catid=70:tutorial-basico-programador-web-php-desde-cero&Itemid=193

[13] http://www.clubdelphi.com/foros/showthread.php?t=37127

[14] http://www.raspberry-projects.com/pi/programming-in-c/mysql/accessing-the-database

Parte II ESTUDIO ECONÓMICO

Capítulo 1 ESTUDIO ECONÓMICO

En este apartado vamos a estudiar los costes asociados del proyecto y el coste total.

Dispositivos	Horas	Precio €	Cantidad	Amortización anual	Subtotal €
Raspberry Pi	1000	39.9	1	25%	250
Router	1000	22	1	25%	5500

Honorario: Mano de obra	Horas	Precio €			Subtotal €
Programador	190	60			11400

Coste Ejecución sin IVA	29025
Coste Ejecución con IVA (21%)	35120,25
Coste total	37227,465

- Al coste total de 37227.465 € se le han incluido los honorarios correspondientes al ingeniero (6%), resultado a partir de los costes de ejecución con IVA (21%), 35120.25 €.

Parte III MANUAL DE USUARIO

1.1 MANUAL DE USUARIO APLICACIÓN WEB

1.1.1 DISEÑO

El diseño del sitio web cuenta con 4 secciones o áreas de información:

• Menú superior: Siempre visible.

• Menú derecho: Solo visible en la sección planta.

• Área de contenidos: Es el espacio principal de trabajo.

• Pie de página: Siempre visible.

La navegación por las distintas secciones del sitio web pretende ser intuitiva.

1.1.2 DESCRIPCIÓN DE CONTENIDOS

Planta

Área de contenido: Representación de la planta de una vivienda con situación actual de los radiadores.

Menú derecho: Diferentes interruptores correspondientes a cada una de las habitaciones, ejemplo: Habitación1: Apagado o encendido del radiador 1.

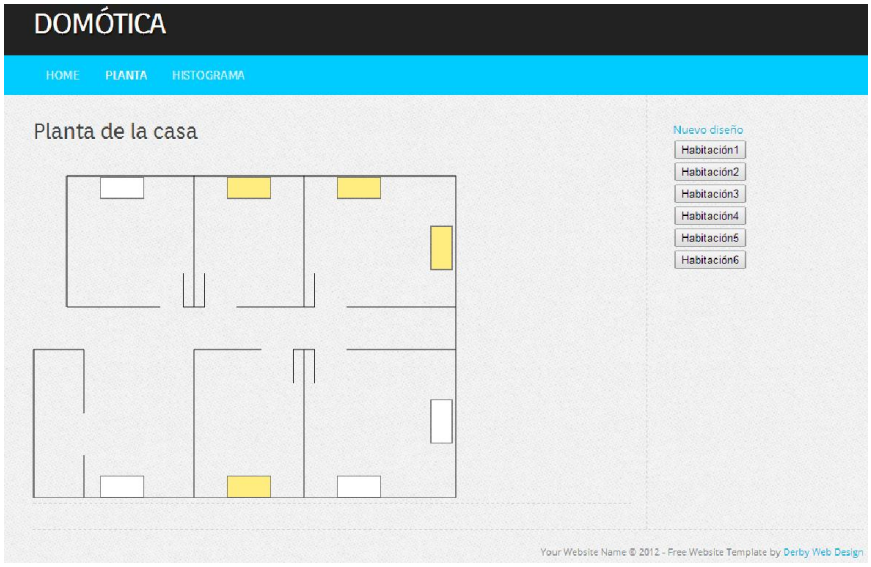

Figura 17. Planta

Histograma

Área de contenido: Acceso a los diferentes histogramas de temperaturas al clickear.

Figura 18. Histograma

Parte IV CÓDIGO FUENTE

Capítulo 1 CÓDIGO FUENTE

El código incluido en esta sección esta referenciado según el siguiente esquema:

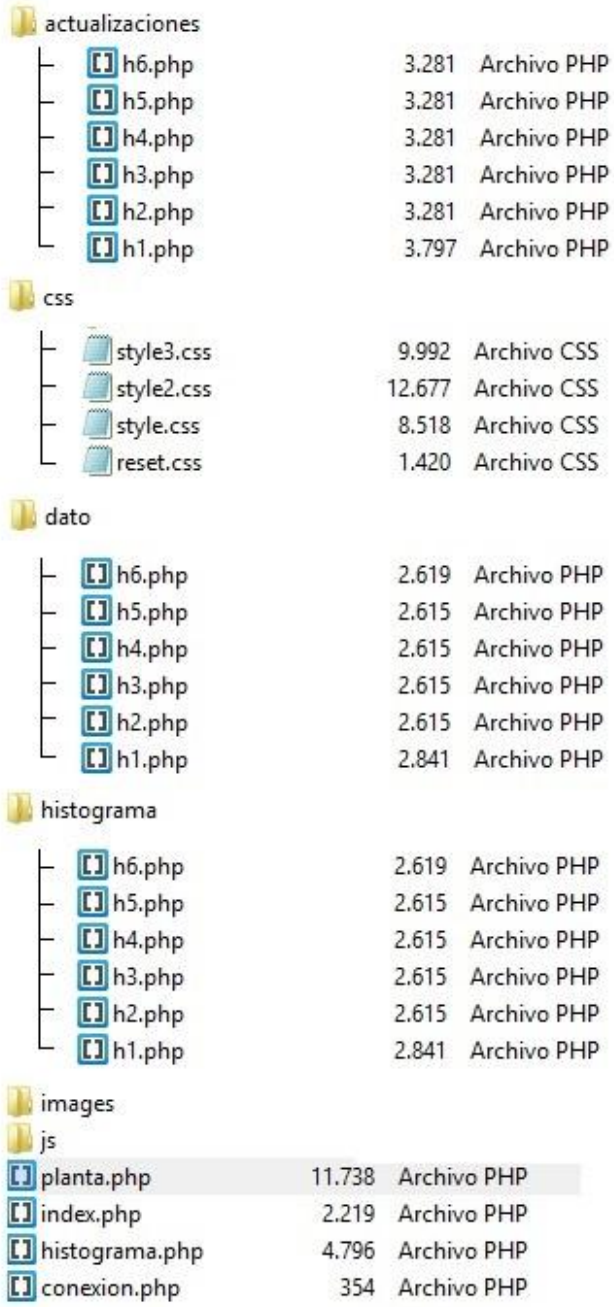

Figura 19. Estructura archivos de la carpeta /www.

index.php

```
<!DOCTYPE html>
<html>
    <head>
        <title>Domótica</title>

        <!--Codificación UTF-8-->
        <meta charset="UTF-8">
        <!--css-->
        <meta name="description" content="">
        <meta name="keywords" content="">
        <meta name="viewport" content="width=device-width,initial-
scale=1">
        <link rel="stylesheet" href="css/style.css">
        <!--JQuery-->
        <script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.
js"></script>
        <script>window.jQuery || document.write('<script
src="js/jquery.js"><\/script>')</script>

        <!--Java-->
        <script src="js/scripts.js"></script>


    </head>
    <body>
         <div class="container darkbg clearfix">

            <div id="headercont" class="clearfix">
                <div id="headerlogo">
                     <h1><a title=""
href="index.php">DOMÓTICA</a></h1>
                </div>
                <div id="headerright">

                </div>
            </div>
        </div>
        <div class="container lightbluebg clearfix">
            <div id="menucont" class="clearfix">
                <div class="menu">
                    <ul>
                        <!--Pestañas-->
                        <li class="active"><a title=""
href="#">Home</a></li>
                        <li><a title=""
href="planta.php">Planta</a></li>
                        <li><a title=""
href="histograma.php">Histograma</a></li>


                    </ul>
                </div>
```

```
            </div>
        </div>
        <div class="container clearfix">
            <div id="maincont" class="clearfix">
                <div id="mainleft">
                    <h1>Información sobre la casa</h1>
                    <?php
                    include("dato/h1.php");
                    $h1=mysql_result($resultado1,0,"TempHoy");
                    echo $h1;
                    ?>
                </div>
            </div>
        </div>
    </div>
    <div class="container clearfix">
        <div id="footercont" class="clearfix">
            <p>Domotica ICAI &copy; 2014 - Free Website by
Alberto</p>
        </div>
    </div>
</body>
</html>
```

planta.php

```
<!DOCTYPE html>
<html>
    <?php

        include("conexion.php");     //Archivo necesario para
conexión
        $consultarad="SELECT * FROM $tabla2";    //Consulta a
RADIADORES
        $resultadorad=mysql_db_query
($base_datos,$consultarad,$conexion);

        //Datos de la tabla RADIADORES
        $e1= mysql_result($resultadorad,0,"Estado");
        $e2= mysql_result($resultadorad,1,"Estado");
        $e3= mysql_result($resultadorad,2,"Estado");
        $e4= mysql_result($resultadorad,3,"Estado");
        $e5= mysql_result($resultadorad,4,"Estado");
        $e6= mysql_result($resultadorad,5,"Estado");
        mysql_free_result($resultadorad);
        mysql_close($conexion);
    ?>
    <head>
        <title>Domótica</title>

        <!--Codificación UTF-8-->
        <meta charset="UTF-8">
        <!--css-->
        <meta name="description" content="">
```

```
        <meta name="keywords" content="">
        <meta name="viewport" content="width=device-width,initial-
scale=1">
        <link rel="stylesheet" href="css/style.css">
        <!--JQuery-->
        <script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.
js"></script>
        <script>window.jQuery || document.write('<script
src="js/jquery.js"><\/script>')</script>

        <!--Java-->
        <script src="js/scripts.js"></script>

        <!--INFORMACIÓN CANVAS-->

        <script>
            //Variables
            var original =0;
            var sit1=<?php  echo $e1;?>;
            var sit2=<?php  echo $e2;?>;
            var sit3=<?php  echo $e3;?>;
            var sit4=<?php  echo $e4;?>;
            var sit5=<?php  echo $e5;?>;
            var sit6=<?php  echo $e6;?>;

            //función para cargar el contexto de un canvas
            function cargaContextoCanvas(idCanvas){
                var elemento = document.getElementById(idCanvas);
                if(elemento && elemento.getContext){
                    var contexto = elemento.getContext('2d');
                    //Comprobación
                    if(contexto){
                        return contexto;
                    }
                }
                return FALSE;
            }
            window.onload = function()
            {
                //Recibimos el elemento canvas
                var ctx = cargaContextoCanvas('micanvas');
                if(ctx)
                {

                    //--PAREDES Y PUERTAS
                    ctx.beginPath();
                    //Desplazamiento
                    ctx.moveTo(40,20);
                    //Dibujar segmento
                    ctx.lineTo(40,175);
                    ctx.lineTo(150,175);
                    ctx.moveTo(40,20);
                    ctx.lineTo(190,20);
                    ctx.lineTo(190,175);
```

```
ctx.moveTo(177.5,175);
ctx.lineTo(202.5,175);
ctx.lineTo(202.5,135);
ctx.moveTo(177.5,175);
ctx.lineTo(177.5,135);

ctx.moveTo(190,20);
ctx.lineTo(320,20);
ctx.lineTo(320,175);
ctx.lineTo(240,175);

ctx.moveTo(320,175);
ctx.lineTo(332.5,175);
ctx.lineTo(332.5,135);

ctx.moveTo(320,20);
ctx.lineTo(499,20);
ctx.lineTo(499,175);
ctx.lineTo(370,175);

ctx.moveTo(499,175);
ctx.lineTo(499,225);
ctx.lineTo(370,225);

ctx.moveTo(499,225);
ctx.lineTo(499,400);
ctx.lineTo(320,400);
ctx.lineTo(320,225);
ctx.lineTo(332.5,225);
ctx.lineTo(332.5,265);

ctx.moveTo(320,225);
ctx.lineTo(307.5,225);
ctx.lineTo(307.5,265);

ctx.moveTo(320,400);
ctx.lineTo(190,400);
ctx.lineTo(190,225);
ctx.lineTo(270,225);

ctx.moveTo(190,400);
ctx.lineTo(60,400);
ctx.lineTo(60,350);

ctx.moveTo(60,400);
ctx.lineTo(1,400);
ctx.lineTo(1,225);
ctx.lineTo(60,225);
ctx.lineTo(60,300);

//--RADIADORES

//Bordes negros de cada radiador
ctx.strokeRect( 80, 22, 50,24);

ctx.strokeRect(230, 22, 50,24);
```

```
                    ctx.strokeRect(360, 22, 50,24);
                    ctx.strokeRect(470, 80, 24,50);

                    ctx.strokeRect(360, 375, 50,24);
                    ctx.strokeRect( 470, 285, 24,50);

                    ctx.strokeRect(230, 375, 50,24);

                    ctx.strokeRect( 80, 375, 50,24);

                    //Coloreado inicial
                    if(original == 0)
                        {
                            //Radiador1
                            if(sit1==1){
                                ctx.fillStyle = '#ffee7f';
//Colorea de amarillo
                                ctx.fillRect(80,22,50,24);
                            }
                            else if(sit1==0){
                                ctx.fillStyle = 'white';
//Colorea de blanco
                                ctx.fillRect(80,22,50,24);
                            }
                            //Radiador2
                            if(sit2==1){
                                ctx.fillStyle = '#ffee7f';
                                ctx.fillRect(230, 22, 50,24);
                            }
                            else if(sit2==0){
                                ctx.fillStyle = 'white';
                                ctx.fillRect(230, 22, 50,24);
                            }
                            //Radiador3
                            if(sit3==1){
                                ctx.fillStyle = '#ffee7f';
                                ctx.fillRect(360, 22, 50,24);
                                ctx.fillRect(470, 80, 24,50);
                            }
                            else if(sit3==0){
                                ctx.fillStyle = 'white';
                                ctx.fillRect(360, 22, 50,24);
                                ctx.fillRect(470, 80, 24,50);
                            }
                            //Radiador4
                            if(sit4==1){
                                ctx.fillStyle = '#ffee7f';
                                ctx.fillRect(360, 375, 50,24);
                                ctx.fillRect(470, 285, 24,50);
                            }
                            else if(sit4==0){
                                ctx.fillStyle = 'white';
                                ctx.fillRect(360, 375, 50,24);
                                ctx.fillRect(470, 285, 24,50);
                            }
```

```
                                //Radiador5
                                if(sit5==1){
                                    ctx.fillStyle = '#ffee7f';
                                    ctx.fillRect(230, 375, 50,24);
                                }
                                else if(sit5==0){
                                    ctx.fillStyle = 'white';
                                    ctx.fillRect(230, 375, 50,24);
                                }
                                //Radiador6
                                if(sit6==1){
                                    ctx.fillStyle = '#ffee7f';
                                    ctx.fillRect(80, 375, 50,24);
                                }
                                else if(sit6==0){
                                    ctx.fillStyle = 'white';
                                    ctx.fillRect(80, 375, 50,24);
                                }
                                original = 1;
                            }
                        ctx.stroke();
                    }//if ctx
                }//windows onload
            </script>
            <!--INFORMACIÓN CANVAS-->
</head>
<body>
        <div class="container darkbg clearfix">

                <div id="headercont" class="clearfix">
                    <div id="headerlogo">
                        <h1><a title=""
href="index.php">DOMÓTICA</a></h1>
                    </div>
                    <div id="headerright">
                    </div>
                </div>
            </div>
            <div class="container lightbluebg clearfix">
                <div id="menucont" class="clearfix">
                    <div class="menu">
                        <ul>
                            <li><a title=""
href="index.php">Home</a></li>
                            <li class="active"><a title=""
href="#">Planta</a></li>
                            <li><a title=""
href="histograma.php">Histograma</a></li>
                        </ul>
                    </div>
                </div>
            </div>
            <div class="container clearfix">
                <div id="maincont" class="clearfix">
                    <div id="mainleft">
                        <h1>Planta de la casa</h1>
```

```
                    <canvas id="micanvas" width="500"
height="400">Recicla tu browser, POR FAVOR!!</canvas>
                    <hr/>
                </div>
                <div id="mainright">
                    <a href="nuevo/nuevo.php" > Nuevo diseño</a>

                <form action="actualizaciones/h1.php"
method="post">
                    <button class="mult_submit" type="submit"
name="submit_mult" value="Modificar"
title="Radiador1">Habitación1</button>
                </form>

                <form action="actualizaciones/h2.php"
method="post">
                    <button class="mult_submit" type="submit"
name="submit_mult" value="Modificar"
title="Radiador2">Habitación2</button>
                </form>

                <form action="actualizaciones/h3.php"
method="post">
                    <button class="mult_submit" type="submit"
name="submit_mult" value="Modificar"
title="Radiador3">Habitación3</button>
                </form>

                <form action="actualizaciones/h4.php"
method="post">
                    <button class="mult_submit" type="submit"
name="submit_mult" value="Modificar"
title="Radiador4">Habitación4</button>
                </form>

                <form action="actualizaciones/h5.php"
method="post">
                    <button class="mult_submit" type="submit"
name="submit_mult" value="Modificar"
title="Radiador5">Habitación5</button>
                </form>

                <form action="actualizaciones/h6.php"
method="post">
                    <button class="mult_submit" type="submit"
name="submit_mult" value="Modificar"
title="Radiador6">Habitación6</button>
                </form>
                </div>
            </div>
        </div>
    </div>
    <div class="container clearfix">

        <div id="footercont" class="clearfix">
```

```
            <p>Your Website Name &copy; 2012 - Free Website
Template by <a title="derby website design" rel="external">Derby
Web Design</a></p>
        </div>
    </div>

</body>
```

histograma.php

```
<!DOCTYPE html>
<html>
    <head>
        <title>Domótica</title>

        <!--Codificación UTF-8-->
        <meta charset="UTF-8">
        <!--css-->
        <meta name="description" content="">
        <meta name="keywords" content="">
        <meta name="viewport" content="width=device-width,initial-
scale=1">
        <link rel="stylesheet" href="css/style.css">
        <!--JQuery-->
        <script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.
js"></script>
        <script>window.jQuery || document.write('<script
src="js/jquery.js"><\/script>')</script>

        <!--Java-->
        <script src="js/scripts.js"></script>


    </head>
    <body>

        <div class="container darkbg clearfix">
            <div id="headercont" class="clearfix">
                <div id="headerlogo">
                    <h1><a title=""
href="index.php">DOMÓTICA</a></h1>
                </div>
                <div id="headerright">

                </div>
            </div>
        </div>
        <div class="container lightbluebg clearfix">
            <div id="menucont" class="clearfix">
                <div class="menu">
                    <ul>
```

```
                        <li><a title=""
href="index.php">Home</a></li>
                        <!--Pestañas-->
                        <li><a title=""
href="planta.php">Planta</a></li>
                        <li class="active"><a title=""
href="#">Histograma</a></li>

                    </ul>
                </div>
            </div>
        </div>
        <div class="container clearfix">
            <div id="maincont" class="clearfix">
                <h1>Histograma</h1>
                <nav class="h_menu">
                    <!--Registro de termperaturas de cada
habitación-->
                    <ul>
                        <li>
                            <a class="h_menu" onclick=" var
ifreim1 = document.getElementById('ifreim1');
ifreim1.style.display = ifreim1.style.display == 'none' ? 'block'
: 'none'; "/>Habitación1</a>
                            <iframe src="histograma/h1.php"
id="ifreim1" style="display:none" width=100% height=250
display=none></iframe>
                        </li>

                        <li>
                            <a class="h_menu" onclick=" var
ifreim2 = document.getElementById('ifreim2');
ifreim2.style.display = ifreim2.style.display == 'none' ? 'block'
: 'none'; "/>Habitación2</a>
                            <iframe src="histograma/h2.php"
id="ifreim2" style="display:none" width=100% height=250
display=none></iframe>
                        </li>

                        <li>
                            <a class="h_menu" onclick=" var
ifreim3 = document.getElementById('ifreim3');
ifreim3.style.display = ifreim3.style.display == 'none' ? 'block'
: 'none'; "/>Habitación3</a>
                            <iframe src="histograma/h3.php"
id="ifreim3" style="display:none" width=100% height=250
display=none></iframe>
                        </li>

                        <li>
                            <a class="h_menu" onclick=" var
ifreim4 = document.getElementById('ifreim4');
ifreim4.style.display = ifreim4.style.display == 'none' ? 'block'
: 'none'; "/>Habitación4</a>
```

```
                                <iframe src="histograma/h4.php"
id="ifreim4" style="display:none" width=100% height=250
display=none></iframe>
                            </li>

                            <li>
                                <a class="h_menu" onclick=" var
ifreim5 = document.getElementById('ifreim5');
ifreim5.style.display = ifreim5.style.display == 'none' ? 'block'
: 'none'; "/>Habitación5</a>
                                <iframe src="histograma/h5.php"
id="ifreim5" style="display:none" width=100% height=250
display=none></iframe>
                            </li>

                            <li>
                                <a class="h_menu" onclick=" var
ifreim6 = document.getElementById('ifreim6');
ifreim6.style.display = ifreim6.style.display == 'none' ? 'block'
: 'none'; "/>Habitación6</a>
                                <iframe src="histograma/h6.php"
id="ifreim6" style="display:none" width=100% height=250
display=none></iframe>
                            </li>
                        </ul>
                    </nav>
                </div>
            </div>
        </div>
    </div>
</div>
<div class="container clearfix">
    <div id="footercont" class="clearfix">
        <p>Domotica ICAI &copy; 2014 - Free Website by Alberto</p>
    </div>
</div>
</body>
</html>
```

conexion.php

```
<?php
    // -- Declaracion de variables PHP y de acceso a BBDD
    $base_datos = "temperatura";
    $tabla2 = "radiadores";
    $direccion_bd = "localhost";
    $usu_bd  = "root";
    $pass_bd = "soanall";
    // Conectamos con BBDD
    $conexion = mysql_connect ($direccion_bd,$usu_bd,$pass_bd) or
die ("No es posible acceder a la BBDD");
?>
```

dato/h1.php

```
<?php
    include("../conexion.php");
    $consulta1="SELECT * FROM habitacion1";
    $resultado1=mysql_db_query ($base_datos,$consulta1,$conexion);

    $h1=mysql_result($resultado1,0,"TempHoy");
    $h2=mysql_result($resultado1,1,"TempHoy");
    $h3=mysql_result($resultado1,2,"TempHoy");
    $h4=mysql_result($resultado1,3,"TempHoy");
    $h5=mysql_result($resultado1,4,"TempHoy");
    $h6=mysql_result($resultado1,5,"TempHoy");
    $h7=mysql_result($resultado1,6,"TempHoy");
    $h8=mysql_result($resultado1,7,"TempHoy");
    $h9=mysql_result($resultado1,8,"TempHoy");
    $h10=mysql_result($resultado1,9,"TempHoy");
    $h11=mysql_result($resultado1,10,"TempHoy");
    $h12=mysql_result($resultado1,11,"TempHoy");
    $h13=mysql_result($resultado1,12,"TempHoy");
    $h14=mysql_result($resultado1,13,"TempHoy");
    $h15=mysql_result($resultado1,14,"TempHoy");
    $h16=mysql_result($resultado1,15,"TempHoy");
    $h17=mysql_result($resultado1,16,"TempHoy");
    $h18=mysql_result($resultado1,17,"TempHoy");
    $h19=mysql_result($resultado1,18,"TempHoy");
    $h20=mysql_result($resultado1,19,"TempHoy");
    $h21=mysql_result($resultado1,20,"TempHoy");
    $h22=mysql_result($resultado1,21,"TempHoy");
    $h23=mysql_result($resultado1,22,"TempHoy");
    $h24=mysql_result($resultado1,23,"TempHoy");

    $a1=mysql_result($resultado1,0,"TempAyer");
    $a2=mysql_result($resultado1,1,"TempAyer");
    $a3=mysql_result($resultado1,2,"TempAyer");
    $a4=mysql_result($resultado1,3,"TempAyer");
    $a5=mysql_result($resultado1,4,"TempAyer");
    $a6=mysql_result($resultado1,5,"TempAyer");
    $a7=mysql_result($resultado1,6,"TempAyer");
    $a8=mysql_result($resultado1,7,"TempAyer");
    $a9=mysql_result($resultado1,8,"TempAyer");
    $a10=mysql_result($resultado1,9,"TempAyer");
    $a11=mysql_result($resultado1,10,"TempAyer");
    $a12=mysql_result($resultado1,11,"TempAyer");
    $a13=mysql_result($resultado1,12,"TempAyer");
    $a14=mysql_result($resultado1,13,"TempAyer");
    $a15=mysql_result($resultado1,14,"TempAyer");
    $a16=mysql_result($resultado1,15,"TempAyer");
    $a17=mysql_result($resultado1,16,"TempAyer");
    $a18=mysql_result($resultado1,17,"TempAyer");
    $a19=mysql_result($resultado1,18,"TempAyer");
    $a20=mysql_result($resultado1,19,"TempAyer");
    $a21=mysql_result($resultado1,20,"TempAyer");
    $a22=mysql_result($resultado1,21,"TempAyer");
    $a23=mysql_result($resultado1,22,"TempAyer");
    $a24=mysql_result($resultado1,23,"TempAyer");
?>
```

actualizaciones/h1.php

```
<!DOCTYPE html>
<html>
    <head>
        <title>h1</title>
        <!--Despues de 5 segundo redireccionamiento a planta.php
para ver cambios realizados-->
        <meta http-equiv="Refresh" content="5; url=../planta.php"
/>
        <meta charset="UTF-8">
        <meta name="description" content="">
        <meta name="keywords" content="">
        <meta name="viewport" content="width=device-width,initial-
scale=1">
        <link rel="stylesheet" href="css/style.css">
        <script>window.jQuery || document.write('<script
src="js/jquery.js"><\/script>')</script>
        <script src="js/scripts.js"></script>
    </head>
    <body>
        <div class="container darkbg clearfix">
            <div id="headercont" class="clearfix">
                <div id="headerlogo">
                     <h1><a title=""
href="index.php">DOMÓTICA</a></h1>
                </div>
                <div id="headerright">

                </div>
            </div>
        </div>
        <div class="container lightbluebg clearfix">
            <div id="menucont" class="clearfix">
                <div class="menu">
                    <ul>
                        <li class="active"><a title=""
href="../index.php">Home</a></li>
                        <li><a title=""
href="../planta.php">Planta</a></li>
                        <li><a title=""
href="../histograma.php">Histograma</a></li>


                    </ul>
                </div>
            </div>
        </div>
        <div class="container clearfix">
            <div id="maincont" class="clearfix">
                <div id="mainleft">
                    <h1>Actualización</h1>
                    <?php
                        //Establecemos coneción
```

```
                        include("../conexion.php");
                        //Consulta tabla RADIADORES
                        $consultarad="SELECT * FROM $tabla2";
                        $resultadorad=mysql_db_query
($base_datos,$consultarad,$conexion);

                        //Estado del radiador 1
                        $r1=
mysql_result($resultadorad,0,"Estado");
                        $r1= intval($r1);

                        if($r1 == 0){
                            echo "Radiador 1 apagado, se
encendera<br></br>";
                            //Seleccionamos la base de datos.

mysql_select_db("temperatura",$conexion);
                            //Actualizamos nuevo valor ENCENDIDO
                            $ssql = "UPDATE radiadores SET
Estado=1 WHERE id=1";
                        }
                        if($r1 == 1){
                            echo "Radiador 1 encendido, se
apagara<br></br>";
                            //Seleccionamos la base de datos.

mysql_select_db("temperatura",$conexion);
                            //Actualizamos nuevo valor APAGADO
                            $ssql = "UPDATE radiadores SET
Estado=0 WHERE id=1";
                        }
                        //Comprobamos actualización
                        if (mysql_query($ssql)) {
                        echo "Actualizacion exitosa ";
                        } else {
                        echo "Error de actualizacion ";
                        }
                        mysql_close($conexion);
                    ?>
                </div>
            </div>
        </div>
    </div>
    <div class="container clearfix">
        <div id="footercont" class="clearfix">
            <p>Domotica ICAI &copy; 2014 - Free Website by
Alberto</p>
        </div>
    </div>
</body>
</html>
```

histograma/h1.php

```
<html>
<?php
    include("../dato/h1.php");
?>

<head>
    <!--JAVA. REPOSITORIO APLICACION GOOGLE-->
    <script type="text/javascript"
src="https://www.google.com/jsapi"></script>
    <!--INFORMACIÓN HISTOGRAMA HABITACIÓN 1-->
    <script type="text/javascript">
        google.load("visualization", "1",
{packages:["corechart"]});
        google.setOnLoadCallback(drawChart);

      function drawChart() {
          var data = google.visualization.arrayToDataTable([
            ['Hour', 'Ayer', 'Hoy'],
            ['00:00', <?php echo $a1;?>,<?php echo $h1;?>],
            ['01:00', <?php echo $a2;?>,<?php echo $h2;?>],
            ['02:00', <?php echo $a3;?>,<?php echo $h3;?>],
            ['03:00', <?php echo $a4;?>,<?php echo $h4;?>],
            ['04:00', <?php echo $a5;?>,<?php echo $h5;?>],
            ['05:00', <?php echo $a6;?>,<?php echo $h6;?>],
            ['06:00', <?php echo $a7;?>,<?php echo $h7;?>],
            ['07:00', <?php echo $a8;?>,<?php echo $h8;?>],
            ['08:00', <?php echo $a9;?>,<?php echo $h9;?>],
            ['09:00', <?php echo $a10;?>,<?php echo $h10;?>],
            ['10:00', <?php echo $a11;?>,<?php echo $h11;?>],
            ['11:00', <?php echo $a12;?>,<?php echo $h12;?>],
            ['12:00', <?php echo $a13;?>,<?php echo $h13;?>],
            ['13:00', <?php echo $a14;?>,<?php echo $h14;?>],
            ['14:00', <?php echo $a15;?>,<?php echo $h15;?>],
            ['15:00', <?php echo $a16;?>,<?php echo $h16;?>],
            ['16:00', <?php echo $a17;?>,<?php echo $h17;?>],
            ['17:00', <?php echo $a18;?>,<?php echo $h18;?>],
            ['18:00', <?php echo $a19;?>,<?php echo $h19;?>],
            ['19:00', <?php echo $a20;?>,<?php echo $h20;?>],
            ['20:00', <?php echo $a21;?>,<?php echo $h21;?>],
            ['21:00', <?php echo $a22;?>,<?php echo $h22;?>],
            ['22:00', <?php echo $a23;?>,<?php echo $h23;?>],
            ['23:00', <?php echo $a24;?>,<?php echo $h24;?>]
        ]);

        var options = {
            title: 'Registro de temperaturas',
            curveType: 'function',
        };

        var chart = new
google.visualization.LineChart(document.getElementById('chart_div'
));
        chart.draw(data, options);
      }
    </script>
```

```
  </head>
  <body>
    <div id="chart_div" style="width: 600px; height:
200px;"></div>
  </body>
</html>
```

Parte V ANEXOS

Capítulo 1 CONOCIMIENTOS DE PROGRAMACIÓN

Adquirir conocimientos de programación realizando cursos vía internet y con libros de programación.

1.1 HTML

Definición: HTML siglas de HyperText Markup Language («lenguaje de marcas de hipertexto»), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, etc.

Aprendizaje: Mediante la aplicación web interactiva Codecademy, realizando un aprendizaje en paralelo con CSS y con "El gran libro de HTML5, CSS3 Y Javasript".

1.2 CSS

Definición: Hojas de Estilo en Cascada (Cascading Style Sheets) es el lenguaje de hojas de estilo utilizado para describir el aspecto y el formato de un documento escrito en un lenguaje de marcas, esto incluye varios lenguajes basados en XML como son XHTML o SVG.

Aprendizaje: Mediante la aplicación web interactiva Codecademy, realizando un aprendizaje en paralelo con HTML y con "El gran libro de HTML5, CSS3 Y Javasript".

1.3 CANVAS

Definición: Canvas (lienzo en inglés) es un elemento HTML incorporado en HTML5 que permite la generación de gráficos dinámicamente por medio del scripting. Permite generar gráficos estáticos y animaciones.

Aprendizaje: A partir de un manual desarrollado por la página web "www.desarrolloweb.com".

1.4 PHP

Definición: PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante.

Aprendizaje: Mediante la aplicación web interactiva Codecademy.